

Черкаський державний технологічний університет
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису

Міхав Володимир Володимирович

УДК 004.738+004.942

ДИСЕРТАЦІЯ

МОДЕЛЬ ТА МЕТОДИ ЗБОРУ І ОБРОБКИ ДАНИХ ДЛЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ У PEER-TO-PEER КОМП'ЮТЕРНИХ МЕРЕЖАХ

123 Комп'ютерна інженерія

12 Інформаційні технології

Подається на здобуття ступеня доктора філософії

Дисертація містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів
мають посилання на відповідне джерело

_____ В. В. Міхав

Наукові керівники:

Мелешко Єлизавета Владиславівна
доктор технічних наук, професор

Лавданський Артем Олександрович
кандидат технічних наук, доцент

Черкаси – 2023

АНОТАЦІЯ

Міхав В. В. Модель та методи збору і обробки даних для рекомендаційних систем у peer-to-peer комп'ютерних мережах. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 123 Комп'ютерна інженерія. – Черкаський державний технологічний університет, Черкаси, 2023.

Актуальність дослідження зумовлена зростанням кількості неструктурованих даних у комп'ютерних мережах різних типів. Для покращення пошуку у великих масивах інформації все частіше звичайні методи пошуку доповнюють рекомендаційними системами. Рекомендаційні системи дозволяють полегшити пошук при великій кількості контенту, доповнюючи або замінюючи класичну пошукову видачу рекомендаціями. Зокрема, вони дозволяють полегшити пошук при великій кількості об'єктів у системі, доповнюючи класичну пошукову видачу рекомендаціями, а в деяких ситуаціях навіть замінюють пошук. Також рекомендаційні системи можуть застосовуватися для ранжування результатів класичного пошуку. Таким чином вони можуть різними способами поєднуватися із звичайними пошуковими алгоритмами. В P2P мережах застосування рекомендаційних систем може мати додаткову користь. Якщо користувач шукає конкретний файл, доданий до мережі раніше, і файл не знайдено з різних причин, можна надати користувачу список рекомендацій з врахуванням його вподобань і, можливо, пошукового запиту. У децентралізованих P2P мережах часто виникає проблема індексації та пошуку файлів на різних пристроях мережі.

P2P комп'ютерні мережі знову актуальні й широко використовуються у наш час завдяки численним перевагам, які вони пропонують. Якщо спочатку вони набули популярності завдяки файлообмінним сервісам, то зараз їх використовують й у багатьох інших сферах. Найважливішою перевагою P2P мереж є децентралізація. Вони можуть працювати без центрального сервера,

що робить їх стійкішими до відмов та атак. Це особливо корисно в умовах збільшення кількості кіберзагроз. P2P також сприяють розвитку сучасних технологій, таких як блокчейн. Багато криптовалют та додатків для управління цифровими активами базуються на P2P принципах, що забезпечує їх безпеку та надійність. У сфері інтернет-телебачення, стрімінгу та онлайн комп'ютерних ігор P2P також стають все популярнішими. Вони дозволяють роздавати контент без великих централізованих серверів, що ефективно зменшує навантаження на інфраструктуру мережі та забезпечує якість стрімінгу. Тож P2P мережі залишаються актуальними завдяки своїй децентралізованій природі, яка сприяє безпеці та стійкості в умовах зростаючих кіберзагроз, а також вони допомагають у розвитку нових технологій та способів розповсюдження контенту.

З різних причин шукані файли можуть бути недоступні для користувача, навіть якщо вони були додані раніше до системи та проіндексовані. Наприклад, комп'ютери, що містять потрібний файл або таблиці маршрутизації до нього чи його частин, вийшли з мережі, або застосовуються технології побудови P2P мережі з ймовірнісними методами пошуку, що не завжди знаходять далеко розташовані від комп'ютера користувача файли тощо. В одноранговій комп'ютерній мережі вузли – учасники мережі динамічно під'єднуються та від'єднуються, при цьому можуть виникнути проблеми з достовірністю та безпекою даних інформаційної системи мережі, зокрема й рекомендаційної, адже нові вузли, що під'єднуються можуть управлятися зловмисниками або бути ураженими шкідливим програмним забезпеченням. Такі вузли можуть бути запрограмовані на викривлення або перехоплення даних рекомендаційної системи. Тому важливо при здійсненні пошуку і створенні рекомендацій враховувати також достовірність та інформаційну безпеку інформації. При формуванні пошукової видачі та створенні рекомендацій користувачам децентралізованих P2P комп'ютерних мереж також важливо враховувати час

на доступ до інформації та знаходити баланс між точністю та швидкістю отримання результатів. Оскільки інформація та таблиці маршрутизації в децентралізованих мережах зберігаються розподілено, важливо враховувати, що деякі файли користувач може отримати швидше, ніж інші, а також, що інколи час пошуку і завантаження файлу може зробити неактуальним зусилля на отримання доступу до нього. Важливим питанням є також кількість пам'яті, яку можна виділити під дані рекомендаційної системи, адже у децентралізованих P2P мережах їх доведеться зберігати розподілено на комп'ютерах учасників мережі. Вибір методу представлення даних, якими оперує рекомендаційна система, має вагомий вплив, оскільки ефективний спосіб представлення даних, необхідних для роботи такої системи, може зменшити кількість потрібних ресурсів та збільшити кількість доступних алгоритмів для формування списків рекомендацій.

Тож рекомендаційні системи значним чином впливають на те, яким користувачі сприймають інформаційний простір, а їх інформаційна безпека та часова і просторова складність алгоритмів є важливими складниками забезпечення якості та ефективності роботи систем пошуку і фільтрації даних у комп'ютерних мережах, особливо децентралізованих.

На основі проведеного дослідження можливих шляхів вирішення наявної науково-практичної задачі було наступним чином сформульовано мету дисертаційної роботи: зменшення витрат часу та пам'яті на роботу рекомендаційної системи в однорангових децентралізованих комп'ютерних мережах при достатній точності створення рекомендацій користувачам.

Проведено дослідження та порівняльний аналіз методів роботи peer-to-peer комп'ютерних мереж, методів зберігання та пошуку даних у них, а також моделей децентралізованих рекомендаційних мереж.

Вперше розроблено математичну модель процесів збору і обробки даних для рекомендаційної системи в одноранговій децентралізованій комп'ютерній мережі, яка відрізняється від відомих можливістю оцінки

ймовірісно-часових характеристик процесів формування і зміни рекомендацій за допомогою мультитригерного GERT-моделювання та врахуванням вимог достовірності і безпеки даних під час змін у структурі мережі, що дозволяє здійснювати раціональний вибір параметрів системи.

Удосконалено метод зберігання даних рекомендаційної системи, який відрізняється від відомих адаптацією до архітектури однорангових децентралізованих комп'ютерних мереж та використанням хеш-таблиць для зберігання даних про користувачів і об'єкти контенту та зв'язних списків для зберігання даних про рекомендації, що дозволило зменшити витрати часу і пам'яті на процеси обробки даних системи.

Удосконалено метод пошуку та фільтрації даних рекомендаційною системою для формування рекомендацій користувачам, який відрізняється від відомих використанням запропонованого методу зберігання даних для збереження проміжних та підсумкових результатів обчислень, що дозволило використовувати його в однорангових децентралізованих комп'ютерних мережах та зменшити витрати часу і пам'яті при забезпеченні високої точності прогнозування вподобань користувачів.

Проведена оцінка якості та ефективності запропонованих моделі та методів шляхом проведення експериментів на програмній імітаційній моделі, а також експериментів із використанням відкритих наборів даних Netflix Prize data та MovieLens.

Практична цінність роботи полягає у такому:

– Розроблена імітаційна модель процесів збору та обробки даних для рекомендаційної системи, що дозволила оцінити ймовірісно-технічні характеристики рекомендаційної системи децентралізованої однорангової комп'ютерної мережі. Отримано аналітичний вираз, за допомогою якого є можливість оцінити щільність розподілу ймовірностей часу ідентифікації стану вузлів децентралізованої рекомендаційної системи. Використання мультитригерного підходу та врахування вимог до достовірності та безпеки

рекомендаційних повідомлень в порівнянні з відомими моделями дозволило підвищити точність результатів моделювання до 5%. Розроблено алгоритми моделювання рекомендаційної системи однорангової децентралізованої комп'ютерної мережі та процесів у ній, що дають можливість проводити тестування різних методів зберігання, пошуку та фільтрації даних у мережі.

– Розроблено алгоритми зберігання даних рекомендаційної системи однорангової децентралізованої комп'ютерної мережі, що дозволяють зменшити витрати пам'яті і часу на процеси зберігання та читання даних системи. Зокрема, розроблена система на основі запропонованих алгоритмів при використанні у децентралізованій одноранговій мережі показує наступні результати в порівнянні з найкращими результатами відомих систем – в 2,4 разів кращі результати по часу заповнення бази даних, в 1,7 разів кращі результати по використаному об'єму пам'яті та в 2,5 разів кращі результати по часу генерації рекомендацій.

– Розроблено алгоритм пошуку та фільтрації даних рекомендаційною системою для формування рекомендацій користувачам в однорангових децентралізованих комп'ютерних мережах, що зменшує витрати часу і пам'яті при забезпеченні високої точності прогнозування вподобань користувачів. Точність розробленого алгоритму сягає до 0,84 в залежності від обраних параметрів системи.

Практичне значення отриманих результатів підтверджено відповідними актами впровадження. Результати дисертаційних досліджень впроваджені і використовуються у діяльності ІТ-компанії ТОВ "ОНІКС-СИСТЕМЗ", а також використано у навчальному процесі Центральноукраїнського національного технічного університету.

Ключові слова: комп'ютерні мережі, рекомендаційні системи, GERT-мережі, структури даних, бази даних, програмне імітаційне моделювання, децентралізовані однорангові комп'ютерні мережі, бінарні діаграми рішень, дерева рішень, зв'язні списки, хеш-таблиці, фільтрація даних, пошук даних.

SUMMARY

Mikhav V. V. Model and methods of data receiving and processing for recommendation systems in peer-to-peer computer networks. – Qualifying scientific work on the rights of the manuscript.

Dissertation for the degree of Doctor of Philosophy (PhD) in the specialty 123 “Computer Engineering”. – Cherkasy State Technological University, Cherkasy, 2023.

The relevance of the research is driven by the increasing volume of unstructured data in various types of computer networks. To enhance search capabilities within vast information repositories, conventional search methods are increasingly complemented by recommendation systems. Recommender systems make it easier to search in a large amount of content by supplementing or even replacing traditional search results with recommendations. In particular, they ease the search process when dealing with an abundance of items within a system supplementing the classic search output with recommendations, and in some cases, even substituting search altogether. Recommendation systems can also be applied to rank classical search results, thus providing various ways to integrate with conventional search algorithms. In P2P networks, the application of recommendation systems can offer additional benefits. If a user is searching for a specific file that was previously shared on the network but cannot be located for various reasons, recommendations can be provided to the user, taking into account their preferences and, perhaps, their search query. In decentralized P2P networks, the problem of indexing and locating files on various network devices often arises.

P2P networks have regained relevance and widespread usage due to their multiple benefits. Though these networks originally gained popularity via file-sharing services, they now serve many other purposes. Decentralization is P2P networks' most important advantage. They can function without a central server, rendering them more resilient to failures and attacks, especially in the current era of

escalating cyber threats. P2P networks also aid in the advancement of contemporary technologies such as blockchain. Numerous cryptocurrencies and digital asset management programs rely on P2P principles, guaranteeing their safety and dependability. In the realm of Internet television, streaming, and online computer games, P2P services are also growing in popularity. Peer-to-peer (P2P) networks enable content distribution without relying on large centralized servers, effectively reducing network infrastructure load and enhancing the quality of streaming. P2P networks remain pertinent due to their decentralized nature, which enhances security and resilience against increasing cyber threats, and contributes to advancing new technologies and content distribution methods.

Files that have been added to the system and indexed may still not be accessible to the user due to various factors. For instance, the computers containing the required file or routing tables may have gone offline, or probabilistic search methods utilized in P2P network technologies may fail to locate files distant from the user's computer. In a peer-to-peer computer network, the nodes of network participants establish and terminate connections dynamically. This process may degrade the reliability and security of network information system data, including the recommendation system. Such problems can occur due to newly connected nodes that are controlled by attackers or affected by malware. These nodes may be programmed to interfere with the data or even cause corruption of the recommendation system. Therefore, it is important to prioritize reliability and information security when searching for and recommending information.

When generating search results and recommendations for users of decentralized P2P computer networks, it is also important to consider the time it takes to access information and strike a balance between accuracy and speed of results. Because information and routing tables are stored in a distributed manner on decentralized networks, it is important to consider that some files can be retrieved faster than others, and that sometimes the time it takes to find and download a file can make access to it irrelevant. Another important issue is the amount of memory

that can be allocated to the recommendation system data, since in a decentralized P2P network it will have to be stored distributed on the computers of the network participants. The choice of how to represent the data used by a recommendation system has a significant impact, since an efficient way of representing the data required by the system can reduce the amount of resources required and increase the number of algorithms available for generating recommendation lists.

Therefore, recommendation systems have a significant impact on how users perceive the information space, and their information security as well as the temporal and spatial complexity of algorithms are important components of ensuring the quality and efficiency of data search and filtering systems in computer networks, especially decentralized ones.

Based on the conducted research on possible ways to address the existing scientific and practical problem, the aim of the dissertation was formulated as follows: to reduce the time and memory costs of operating a recommendation system in peer-to-peer decentralized computer networks while ensuring sufficient accuracy in generating recommendations for users.

A study and comparative analysis of peer-to-peer computer network operation methods, data storage and retrieval methods within them, as well as models of decentralized recommendation networks, have been conducted.

For the first time, a mathematical model of data collection and processing processes for a recommender system in a peer-to-peer decentralized computer network has been developed. It differs from the known ones by the possibility of assessing the probabilistic and temporal characteristics of the processes of forming and changing recommendations using multi-trigger GERT modeling and taking into account the requirements for data reliability and security during changes in the network structure, which allows to make a rational choice of system parameters.

The method of storing data of the recommender system has been improved, which differs from the known ones by adapting to the architecture of peer-to-peer decentralized computer networks and using hash tables to store data on users and

content objects and linked lists to store data on recommendations, which reduced the time and memory consumption for data processing of the system.

The method of searching and filtering data by a recommender system for generating recommendations to users has been improved. It differs from the known ones by using the proposed method of data storage to store intermediate and final results of calculations, which made it possible to use it in peer-to-peer decentralized computer networks and reduce time and memory costs while ensuring high accuracy of predicting user preferences.

The quality and effectiveness of the proposed model and methods were evaluated by conducting experiments on a software simulation model, as well as experiments using the open datasets Netflix Prize data and MovieLens.

The practical value of the work lies in the following:

- A simulation model of data collection and processing processes for a recommender system has been developed, which has made it possible to evaluate the probabilistic and technical characteristics of a recommender system of a decentralized peer-to-peer computer network. An analytical expression has been obtained that makes it possible to estimate the probability density function of the time of identifying the state of the nodes of a decentralized recommender system. The use of a multi-trigger approach and consideration of the requirements for the reliability and security of recommendation messages in comparison with known models allowed to increase the accuracy of the modeling results by up to 5%. The algorithms for modeling the recommender system of a peer-to-peer decentralized computer network and processes in it have been developed, which make it possible to test various methods of storing, searching and filtering data in the network.

- The algorithms for storing data of the recommender system of a peer-to-peer decentralized computer network have been developed, which allow reducing the memory and time consumption for storing and reading system data. In particular, the developed system based on the proposed algorithms, when used in a decentralized peer-to-peer network, shows the following results compared to the

best results of known systems: 2.4 times better results in terms of database filling time, 1.7 times better results in terms of memory usage, and 2.5 times better results in terms of recommendation generation time.

– An algorithm for searching and filtering data by a recommender system for generating recommendations to users in peer-to-peer decentralized computer networks has been developed, which reduces time and memory costs while ensuring high accuracy in predicting user preferences. The accuracy of the developed algorithm reaches up to 0.84, depending on the selected system parameters.

The practical significance of the results obtained is confirmed by the relevant acts of implementation. The results of the dissertation research have been implemented and are utilized in the activities of the IT company "ONIX-SYSTEMS" LLC and have also been incorporated into the educational process at the Central Ukrainian National Technical University.

Keywords: computer networks, recommender systems, GERT networks, data structures, databases, computer simulation modeling, decentralized peer-to-peer computer networks, binary decision diagrams, decision trees, linked lists, hash tables, data filtering, data search.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

1. Mikhav V., Semenov S., Meleshko Y., Yakymenko M., Shulika Y. Constructng the mathematical model of a recommender system for decentralized peer-to-peer computer networks. *Eastern-European Journal of Enterprise Technologies*. 2023. Vol. 4 (9(124)). P. 24–35. (ISSN 1729-3774). DOI: 10.15587/1729-4061.2023.286187. URL: <https://journals.urau.ua/eejet/article/view/286187> (SCOPUS)
2. Meleshko Ye., Drieieva H., Drieiev O., Yakymenko M., Mikhav V., Shymko S. (2023). A Method of Routing of Fractal-like Traffic with Prediction of Router Load for Reduce the Probability of Network Packet Loss. *Computational Linguistics and Intelligent Systems (COLINS 2023)*: CEUR-WS, Kharkiv, Ukraine / O. Cherednichenko, L. Chyrun, V. Vysotska (Ed.). 2023. Vol. 3403(III). P. 434–448. (ISSN 1613-0073). DOI: 10.5281/zenodo.8231189. URL: <https://ceur-ws.org/Vol-3403/paper34.pdf>. (SCOPUS)
3. Drieieva H., Drieiev O., Meleshko Ye., Yakymenko M., Mikhav V. A method of determining the fractal dimension of network traffic by its probabilistic properties and experimental research of the quality of this method. *Computational Linguistics and Intelligent Systems (COLINS 2022)*: CEUR-WS, Gliwice, Poland / V. Lytvyn (Ed.). 2022. Vol. 3154. P. 1694–1707. (ISSN 1613-0073). DOI: 10.5281/zenodo.7892006. URL: <https://ceur-ws.org/Vol-3171/paper120.pdf>. (SCOPUS)
4. Мелешко Є. В., Дрєєв О. М., Лавданський А. О. Модель рекомендаційної системи для комп'ютерних мереж типу P2P. *Вісник Черкаського державного технологічного університету*. 2023. Вип. 1. С. 52–60. DOI: 10.24025/2306-4412.1.2023.273495. URL: <http://vtn.chdtu.edu.ua/article/view/273495> (фахове видання категорії Б)
5. Міхав В. В., Мелешко Є. В. Метод роботи рекомендаційної системи у комп'ютерній мережі типу peer to peer. *Системи управління,*

навігації та зв'язку. Полтава: ПНТУ, 2023. Т. 1(71). С. 112–117. DOI: 10.26906/SUNZ.2023.1.112. URL: <http://journals.nupp.edu.ua/sunz/article/view/2839> **(фахове видання категорії Б)**

6. Drieieva H., Meleshko Ye., Drieiev O., Mikhav V. Computer simulation model of a computer network with fractal traffic for testing routing algorithms. *Advanced Information Systems*. 2022. No. 6 (4). P. 11–18. DOI: 10.20998/2522-9052.2022.4.02. URL: <http://ais.khpi.edu.ua/article/view/268306>. **(фахове видання категорії Б)**

7. Міхав В. В., Мелешко Є. В., Якименко М. С., Шуліка Я. П. Розробка системи управління базою даних рекомендаційної системи для комп'ютерних та комп'ютерно-інтегрованих систем. *Центральноукраїнський науковий вісник. Технічні науки: зб. наук. пр.* Кропивницький: ЦНТУ, 2022. Вип. 5 (36). С. 130–136. DOI: 10.32515/2664-262X.2022.5(36).2.130-136. URL: [http://mapiea.kntu.kr.ua/pdf/5\(36\)_II/19.pdf](http://mapiea.kntu.kr.ua/pdf/5(36)_II/19.pdf) **(фахове видання категорії Б)**

8. Міхав В. В., Мелешко Є. В., Якименко М. С., Бащенко Д. В. Методи зберігання даних рекомендаційної системи на основі зв'язних списків. *Системи управління, навігації та зв'язку*. Полтава: ПНТУ, 2021. Т. 4(66). С. 59–62. DOI: 10.26906/SUNZ.2021.4.059. URL: <http://journals.nupp.edu.ua/sunz/article/view/2422> **(фахове видання категорії Б)**

9. Міхав В. В., Мелешко Є. В., Якименко М. С. Метод зберігання даних рекомендаційної системи на основі бінарних діаграм рішень. *Системи управління, навігації та зв'язку* Полтава: ПНТУ, 2020. Т. 2 (60). С. 85–89. DOI: 10.26906/SUNZ.2020.2.085. URL: <http://journals.nupp.edu.ua/sunz/article/view/1836> **(фахове видання категорії Б)**

10. Міхав В. В. Програмне забезпечення для моделювання мереж репутації користувачів соціальних веб-ресурсів на основі бінарних діаграм рішень. *Системи управління, навігації та зв'язку*. Полтава: ПНТУ,

2019. Т. 5 (57). С. 78–83. DOI: <https://doi.org/10.26906/SUNZ.2019.5.078>.
URL: <http://journals.nupp.edu.ua/sunz/article/view/1720> **(фахове видання)**

11. Міхав В. В., Мелешко Є. В., Лавданський А. О. Дослідження принципів роботи однорангових децентралізованих структурованих комп'ютерних мереж. *Інформаційна безпека та комп'ютерні технології*: тези доп. VI Міжнар. наук.-практ. конф. (Кропивницький, 20–21 квіт. 2023 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2023. С. 51–52.

12. Міхав В. В., Мелешко Є. В., Дреєв О. М., Миронець І. В. Модель рекомендаційної системи для комп'ютерної мережі типу peer to peer. *Інновації та перспективні шляхи розвитку інформаційних технологій (ІПШРІТ-2022)*: тези доп. Міжнар. наук.-практ. Інтернет-конф., (Черкаси, 9 груд. 2022 р.) / упоряд.: Т. О. Прокопенко, Я. В. Тарасенко. М-во освіти і науки України, Черкас. держ. технол. ун-т, 2022. С. 59–60.

13. Дреєва Г. М., Дреєв О. М., Мелешко Є. В., Якименко М. С., Міхав В. В. Дослідження якості роботи методу визначення фрактальної розмірності мережевого трафіку за його ймовірнісними властивостями. *Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління*: тези доп. 12-ї Міжнар. наук.-техн. конф., (Баку–Харків–Жиліна, 27–28 квіт. 2022 р.) / уклад. В. В. Косенко. Харків: ФОП Петров В. В., 2022. Т. 1. С. 126.

14. Дреєва Г. М., Мелешко Є. В., Міхав В. В. Програмна імітаційна модель комп'ютерної мережі для тестування алгоритмів маршрутизації трафіку. *Автоматика, комп'ютерно-інтегровані технології та проблеми енергоефективності в промисловості і сільському господарстві*: матеріали Міжнар. наук.-техн. конф., (Кропивницький, 10–11 листоп. 2022 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: Ексклюзив-Систем, 2022. С. 44–45.

15. Міхав В. В., Мелешко Є. В., Якименко М. С., Бащенко Д. В. Методи зберігання даних у рекомендаційних системах. *Цифрова трансформація*

суспільства: тези доп. I Міжнар. наук.-практ. конф. (Кропивницький, 21–22 квіт. 2022 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2022. С. 44–45.

16. Міхав В. В., Мелешко Є. В., Якименко М. С. Дослідження можливостей графової бази даних Neo4j для застосування у розробці рекомендаційних систем соціальних мереж. *Комбінаторні конфігурації та їхні застосування*: тези доп. XXIII Міжнар. наук.-практ. семінару ім. А. Я. Петренюка, присвяч. 70-річчю ЛА НАУ (Запоріжжя–Кропивницький, 13–15 трав. 2021 р.) / за ред. Г. П. Донця. Кропивницький: Ексклюзив-Систем, 2021. С. 124–127.

17. Міхав В. В., Мелешко Є. В. Застосування методу delta-encoding для представлення даних рекомендаційної системи. *Інформаційна безпека та комп'ютерні технології*: тези доп. IV Міжнар. наук.-практ. конф. (Кропивницький, 15–16 квіт. 2021 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2021. С. 57.

18. Міхав В. В., Мелешко Є. В. Порівняння стратегій редагування бінарних діаграм рішень для роботи з графовими даними. *Інформація, комунікація, суспільство*: тези доп. 9-ої Міжнар. наук. конф. (Львів, 21–23 трав. 2020 р.) / М-во освіти і науки України, Нац. ун-т «Львівська політехніка». Львів: Вид-во Львів. політехніки, 2020. С. 17–18.

19. Міхав В. В., Мелешко Є. В. Метод оптимізації швидкодії бінарних діаграм рішень при представленні даних рекомендаційної системи. *Інформаційна безпека та інформаційні технології*: тези доп. II Міжнар. наук.-практ. конф. (Кропивницький, 2–3 квіт. 2020 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2020. С. 17.

20. Міхав В. В., Мелешко Є. В., Чабан О. О. Способи побудови рекомендаційних систем для соціальних мереж з врахуванням репутації користувачів. *Перспективні напрямки інформаційних і комп'ютерних систем*

та мереж, комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті: тези доп. Міжнар. наук.-практ. конф. (Кропивницький, 13–14 лист. 2019 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2019. С. 86–87.

21. Міхав В. В. Алгоритми динамічного впорядкування змінних у впорядкованих бінарних діаграмах рішень. *Комбінаторні конфігурації та їх застосування: тези доп. XVII Міжнар. наук.-практ. семінару (Кіровоград, 17–18 квіт. 2015 р.) / за ред. Г. П. Донця. Кропивницький: ЦНТУ, 2015. С. 75–78.*

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ	20
ВСТУП.....	23
Список літератури до вступу	31
РОЗДІЛ 1. ДОСЛІДЖЕННЯ МЕТОДІВ РОБОТИ ОДНОРАНГОВИХ КОМП'ЮТЕРНИХ МЕРЕЖ ТА МОЖЛИВОСТІ РОЗРОБКИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ ДЛЯ НИХ.....	40
1.1. Дослідження методів роботи однорангових комп'ютерних мереж	40
1.2. Дослідження можливості розробки рекомендаційних систем для однорангових комп'ютерних мереж	50
1.3. Дослідження методів та структур даних для зберігання інформації рекомендаційної системи в однорангових комп'ютерних мережах.....	56
1.4. Цілі та задачі дисертаційного дослідження.....	61
Висновки до розділу 1	63
Список літератури до розділу 1	64
РОЗДІЛ 2. МАТЕМАТИЧНА МОДЕЛЬ ЗБОРУ І ОБРОБКИ ДАНИХ ДЛЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ДЕЦЕНТРАЛІЗОВАНОЇ ОДНОРАНГОВОЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ.....	72
2.1. Матеріали та методи дослідження для розробки математичної моделі збору і обробки даних для рекомендаційної системи децентралізованої однорангової комп'ютерної мережі	72
2.2. Розробка математичної моделі збору і обробки даних для рекомендаційної системи децентралізованої однорангової комп'ютерної мережі	75
2.2.1. Структура рекомендаційної системи однорангової децентралізованої комп'ютерної мережі з використанням мультитригерного підходу та врахуванням підвищених вимог до достовірності і безпеки даних	75

2.2.2. Формалізація та оцінка ймовірно-часових характеристик процесів формування та зміни рекомендацій в одноранговій децентралізованій комп'ютерній мережі.....	80
2.2.3. Порівняльні дослідження запропонованої GERT-моделі рекомендаційної системи для децентралізованої однорангової комп'ютерної мережі	87
2.2.4. Аналіз результатів та пропозиції щодо подальшого розвитку розробки математичної моделі збору і обробки даних для рекомендаційної системи децентралізованої однорангової комп'ютерної мережі	92
Висновки до розділу 2	94
Список літератури до розділу 2	95
РОЗДІЛ 3. МЕТОД ЗБЕРІГАННЯ ДАНИХ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ, АДАПТОВАНИЙ ДО АРХІТЕКТУРИ ОДНОРАНГОВИХ ДЕЦЕНТРАЛІЗОВАНИХ КОМП'ЮТЕРНИХ МЕРЕЖ.....	98
3.1. Розробка методу програмного імітаційного моделювання рекомендаційної системи для комп'ютерних мереж різних архітектур	99
3.3.1. Представлення орієнтованих графів за допомогою бінарних діаграм рішень у програмній моделі рекомендаційної системи	100
3.3.2. Опис методу програмного імітаційного моделювання рекомендаційної системи	108
3.2. Розробка методу зберігання даних рекомендаційної системи на основі бінарних діаграм рішень.....	111
3.3. Проведення експериментів для дослідження різних стратегій редагування бінарних діаграм рішень у роботі з графовими даними рекомендаційної системи	116
3.4. Проведення експериментів для порівняльного аналізу методів зберігання інформації рекомендаційної системи децентралізованої однорангової комп'ютерної мережі на основі різних структур даних	119

3.5. Розробка методу зберігання даних рекомендаційної системи на основі розгорнутих зв'язних списків та хеш-таблиць.....	126
3.5.1. Проведення експериментів для порівняльного аналізу запропонованого методу зберігання даних рекомендаційної системи з існуючими.....	132
Висновки до розділу 3	134
Список літератури до розділу 3	136
РОЗДІЛ 4. МЕТОД ПОШУКУ ТА ФІЛЬТРАЦІЇ ДАНИХ РЕКОМЕНДАЦІЙНОЮ СИСТЕМОЮ ДЛЯ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ КОРИСТУВАЧАМ ОДНОРАНГОВОЇ ДЕЦЕНТРАЛІЗОВАНОЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ.....	143
4.1. Розробка аналітичної моделі рекомендаційної системи для децентралізованої однорангової комп'ютерної мережі	143
4.2. Розробка способу формування рекомендацій для користувачів у P2P мережах з урахуванням кількості переходів до завантаження даних.....	149
4.3. Розробка методу пошуку та фільтрації даних рекомендаційною системою для формування рекомендацій користувачам однорангової децентралізованої комп'ютерної мережі	154
4.4. Проведення експериментів для визначення якості роботи запропонованого методу пошуку та фільтрації даних рекомендаційною системою однорангової децентралізованої комп'ютерної мережі	158
Висновки до розділу 4	161
Список літератури до розділу 4	162
ОСНОВНІ ВИСНОВКИ.....	165
ДОДАТКИ.....	168
Додаток А. Акти впровадження дисертаційних досліджень	169
Додаток Б. Список публікацій здобувача за темою дисертації та відомості про апробацію результатів дисертації.....	174

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ

BitTorrent – пірінговий (P2P) мережевий протокол для кооперативного обміну файлами через мережу Інтернет.

В-дерево (b-tree) – структура даних, представлена збалансованим та сильно розгалуженим деревом пошуку. Кожен вузол В-дерева, крім листків, є упорядкованим списком, у якому чергуються ключі і вказівники на нащадків. Ключі вузла вказують інтервал, у якому знаходяться ключі нащадку.

Chord – метод маршрутизації в однорангових децентралізованих структурованих комп'ютерних мережах, реалізований на основі розподіленої хеш-таблиці. Chord визначає, як саме ключі призначаються вузлам, і як вузол може виявити значення для цього ключа, спочатку виявивши вузол, відповідальний за цей ключ.

Distributed hash table (DHT) – клас децентралізованих розподілених систем пошуку, що застосовуються у однорангових децентралізованих структурованих комп'ютерних мережах і працюють подібно до хеш-таблиці.

Flooding – метод пошуку ресурсів в однорангових децентралізованих неструктурованих мережах, у якому пошуковий запит передається від сусіда до сусіда, поки ресурс не буде знайдено або поки не буде опитано усі вузли.

GERT-мережі – метод моделювання, що застосовується у випадках організації проекту, коли існує багатоваріантність способів його реалізації.

Gnutella – повністю децентралізована файлообмінна мережа, використовує для роботи однорангову комп'ютерну мережу, побудовану з користувачів мережі Інтернет.

Kademlia – метод маршрутизації в однорангових децентралізованих структурованих комп'ютерних мережах, реалізований на основі розподіленої хеш-таблиці. Протокол Kademlia визначає структуру мережі, що регулює зв'язок між вузлами, та спосіб обміну інформацією в ній.

P2P (peer-to-peer) – однорангові комп'ютерні мережі, які засновані на принципі рівноправності вузлів мережі і характеризуються тим, що будь-які

вузли можуть з'єднуватися між собою для надання послуг один одному, на відміну від традиційної архітектури, у якій лише окрема категорія учасників (сервери), може надавати певні сервіси іншим.

Бінарні діаграми рішень (БДР, BDD) – економна форма представлення булевих функцій у вигляді орієнтованого ациклічного графа. Вершини графа представляють аргументи функції, листки – її двійкові значення. БДР дають можливість зберігати дані у стисненому вигляді та швидко отримувати значення функції за її параметрами, але редагування БДР потребує складних обчислень.

B+-дерево (B+-tree) – різновид B-дерева, що відрізняється тим, що воно зберігає усі значення у листкових вузлах, а листкові вузли мають посилення на сусіда, завдяки чому можна обійти усі значення без обходу всього дерева. Завдяки великій розгалуженості дерева підтримується мала висота дерева, що дозволяє переглядати невеликий об'єм даних за один прохід, а завдяки правилам побудови значення зберігаються у порядку зростання ключа.

Зв'язний список (linked list) – структура даних, у якій кожен елемент містить поля даних та вказівник на наступний елемент. Основна перевага цієї структури полягає у сталому часі додавання нового елемента. Проте для кожного елемента потрібно виділяти новий блок пам'яті, тому менеджер пам'яті спричиняє значні затримки та накладні витрати пам'яті в процесі роботи.

Розгорнутий зв'язний список (unrolled linked list) – зв'язний список, кожен елемент якого містить масив логічних елементів. Це дозволяє об'єднати переваги масивів та зв'язних списків. Об'єднання блоків логічних елементів у список дозволяє додавати нові елементи без зміни розміру блоку пам'яті, економити пам'ять на вказівниках та ефективніше використовувати кеш процесора завдяки послідовному розташуванню елементів. При послідовному заповненні списку гарантується, що незаповненим лишиться не більше одного блоку елементів.

Хеш-таблиця (hash map) – структура даних, у якій пошук елементу здійснюється на основі його ключа. На розташування елемента у хеш-таблиці вказує хеш-значення його ключа. Якщо декілька елементів мають однаковий хеш, то виникає колізія. Існує два методи розв’язання колізій – закрита та відкрита адресації. При закритій адресації кожен елемент таблиці – це зв’язний список, і усі елементи з однаковим хешем додаються до одного списку. При відкритій адресації у випадку колізії обирається нова позиція елементу. Відкрита адресація використовує фіксований об’єм пам’яті і не потребує додаткових вказівників, але для ефективності операцій вставки і пошуку таблиця має бути заповнена не більш ніж наполовину, що спричиняє додаткові витрати пам’яті.

ВСТУП

Актуальність. Актуальність дослідження зумовлена зростанням кількості неструктурованих даних у комп'ютерних мережах різних типів [1, 2]. Для покращення пошуку у великих масивах інформації все частіше звичайні методи пошуку доповнюють рекомендаційними системами [3, 4]. Рекомендаційні системи дозволяють полегшити пошук при великій кількості контенту, доповнюючи або замінюючи класичну пошукову видачу рекомендаціями [3]. Зокрема, вони дозволяють полегшити пошук при великій кількості об'єктів у системі, доповнюючи класичну пошукову видачу рекомендаціями, а в деяких ситуаціях навіть замінюють пошук. Також рекомендаційні системи можуть застосовуватися для ранжування результатів класичного пошуку [4]. Таким чином вони можуть різними способами поєднуватися із звичайними пошуковими алгоритмами. В P2P мережах застосування рекомендаційних систем може мати додаткову користь [5, 6]. Якщо користувач шукає конкретний файл, що був доданий до мережі раніше, і цей файл не було знайдено з різних причин, можна надати користувачу список рекомендацій з врахуванням його вподобань та, можливо, пошукового запиту. У децентралізованих P2P мережах часто виникає проблема індексації та пошуку файлів на різних пристроях мережі.

P2P комп'ютерні мережі знову актуальні і широко використовуються у наш час через безліч переваг, які вони пропонують [7-12]. Якщо спочатку вони набули популярності завдяки файлообмінним сервісам, то зараз їх використовують і у багатьох інших сферах. Найважливішою перевагою P2P мереж є децентралізація. Вони можуть працювати без центрального сервера, що робить їх стійкішими до відмов та атак, що особливо корисно в умовах збільшення кількості кіберзагроз. P2P також сприяють розвитку сучасних технологій, таких як блокчейн. Багато криптовалют та додатків для управління цифровими активами базуються на P2P принципах, що забезпечує їх безпеку та надійність. У сфері інтернет-телебачення, стрімінгу та онлайн комп'ютерних

ігор P2P також стають все популярнішими. Вони дозволяють роздавати контент без великих централізованих серверів, що ефективно зменшує навантаження на інфраструктуру мережі та забезпечує якість стрімінгу. Тож P2P мережі залишаються актуальними завдяки своїй децентралізованій природі, яка сприяє безпеці та стійкості в умовах зростаючих кіберзагроз, а також допомагають в розвитку нових технологій та способів розповсюдження контенту.

З різних причин шукані файли можуть бути недоступні для користувача, навіть якщо вони були додані раніше до системи та проіндексовані [12-14]. Наприклад, комп'ютери, що містять потрібний файл або таблиці маршрутизації до нього чи його частин, вийшли з мережі, або застосовуються технології побудови P2P мережі з ймовірнісними методами пошуку, що не завжди знаходять далеко розташовані від комп'ютера користувача файли тощо. В одноранговій комп'ютерній мережі вузли – учасники мережі динамічно під'єднуються та від'єднуються, при цьому можуть виникнути проблеми з достовірністю та безпекою даних інформаційної системи мережі, зокрема й рекомендаційної, адже нові вузли, що під'єднуються, можуть управлятися зловмисниками або бути ураженими шкідливим програмним забезпеченням. Такі вузли можуть бути запрограмовані на викривлення або перехоплення даних рекомендаційної системи. Тому важливо при здійсненні пошуку і створенні рекомендацій користувачам враховувати також достовірність та інформаційну безпеку інформації.

При формуванні пошукової видачі та створенні рекомендацій користувачам децентралізованих P2P комп'ютерних мереж також важливо враховувати час на доступ до інформації та знаходити баланс між точністю та швидкістю отримання результатів [4, 15]. Оскільки інформація та таблиці маршрутизації в децентралізованих мережах зберігаються розподілено, важливо враховувати, що деякі файли користувач може отримати швидше за інші, а також те, що інколи час пошуку і завантаження файлу може зробити неактуальними зусилля на отримання доступу до нього. Важливим питанням

є також кількість пам'яті, що можна виділити під дані рекомендаційної системи, адже у децентралізованій P2P мережах їх доведеться зберігати розподілено на комп'ютерах учасників мережі. Вибір методу представлення даних, якими оперує рекомендаційна система, має вагомий вплив, оскільки ефективний спосіб представлення даних, необхідних для роботи такої системи, може зменшити кількість потрібних ресурсів та збільшити кількість доступних алгоритмів для формування списків рекомендацій.

Тож рекомендаційні системи значним чином впливають на те, яким користувачі сприймають інформаційний простір, а їх інформаційна безпека та часова і просторова складність алгоритмів є важливими складниками забезпечення якості та ефективності роботи систем пошуку і фільтрації даних у комп'ютерних мережах, особливо децентралізованих.

Найважливіші роботи в галузі рекомендаційних систем були створені наступними зарубіжними і вітчизняними вченими: Річчі Ф. [16], Рокач Л. [16], Шاپіра Б. [16], Кантор П. Б. [16], Берк Р. [17-20], Мобашер Б. [18-20], Вільямс Ч. [18, 20], Фанк С. [21], Пасічник В. В. [22, 23], Артеменко О. І. [22, 23], Стех Ю. В. [24-27], Лобур М. В. [25-27], Верес О. М. [28-29]. Праці наступних науковців внесли значний внесок в розробку рекомендаційних систем для однорангових комп'ютерних мереж та децентралізованих систем: Бобаділля Д. [30], Ліпіцці К. [30], Бадіс Л. [5], Амад М. [5], Айссані Д. [5], Аббар С. [5], Йю Х. [31], Чунг К. [31].

З огляду на зазначене вище важливою **науково-практичною задачею**, яка вирішується у цій роботі, є підвищення ефективності методів обробки даних в однорангових децентралізованих комп'ютерних мережах шляхом розробки і удосконалення моделей і методів роботи рекомендаційних систем для цієї архітектури мереж.

Для рішення обраної науково-практичної задачі було вирішено зменшити витрати часу та пам'яті при роботі рекомендаційної системи на етапах зберігання, пошуку та фільтрації даних однорангової децентралізованої комп'ютерної мережі.

Зв'язок роботи з науковими програмами, планами, темами.

Дисертаційна робота виконана у межах пріоритетних наукових напрямів, які охоплюють актуальні проблеми, відповідно до рішення Президії Національної академії наук України від 30 січня 2019 року «Про Основні наукові напрями та найважливіші проблеми фундаментальних досліджень у галузі природничих, технічних, суспільних і гуманітарних наук Національної академії наук України на 2019–2023 роки», «Інформатика» за темами: «Розроблення обчислювальних алгоритмів і процедур з метою вирішення практичних задач міждисциплінарного характеру для застосувань, що належать до науково-технічної та соціально-економічної сфер діяльності людини», «Розроблення математичних методів та систем моделювання об'єктів та процесів». Дисертаційну роботу виконано у межах зареєстрованих науково-дослідних робіт Центральноукраїнського національного технічного університету: «Моделювання та аналіз складних мереж та інформаційних систем» (ДР № 0119U003587) та «Методи використання інформаційних технологій та інтелектуальних систем для аналізу даних та забезпечення інформаційної безпеки суспільства» (ДР № 0116U008161).

Мета і задачі дослідження. Мета дисертаційної роботи – зменшення витрат часу та пам'яті на роботу рекомендаційної системи в однорангових децентралізованих мережах при достатній точності створення рекомендацій користувачам.

Мета дисертаційної роботи визначає необхідність розв'язання таких **основних задач:**

1. Розробити математичну модель збору і обробки даних для рекомендаційної системи однорангової децентралізованої комп'ютерної мережі.

2. Удосконалити метод зберігання даних рекомендаційної системи з врахуванням особливостей архітектури однорангових децентралізованих комп'ютерних мереж.

3. Удосконалити метод пошуку та фільтрації даних рекомендаційною системою для формування рекомендацій з врахуванням особливостей

архітектури однорангових децентралізованих комп'ютерних мереж.

4. Провести експериментальне дослідження запропонованих методів роботи рекомендаційних систем для однорангових децентралізованих комп'ютерних мереж.

Об'єктом дослідження є процес формування списку рекомендацій користувачам з об'єктів контенту однорангових децентралізованих комп'ютерних мереж.

Предметом дослідження є методи зберігання, пошуку та фільтрації даних в однорангових децентралізованих комп'ютерних мережах.

Методи дослідження. Для вирішення завдань математичного моделювання використано технології GERT-мережевого моделювання, комп'ютерної алгебри та чисельних методів. Для розробки методів зберігання, пошуку і фільтрації даних рекомендаційною системою та реалізації програмного моделювання було використано теорію алгоритмів, теорію структур даних, теорію комп'ютерних мереж, теорію побудови рекомендаційних систем та теорію об'єктно-орієнтованого програмування. Для визначення якості запропонованих методів використано експерименти на програмній імітаційній моделі та на відкритих наборах даних і теорію статистики.

Наукова новизна одержаних результатів полягає у такому:

– *Вперше розроблено математичну модель процесів збору і обробки даних для рекомендаційної системи в одноранговій децентралізованій комп'ютерній мережі, яка відрізняється від відомих можливістю оцінки ймовірнісно-часових характеристик процесів формування та зміни рекомендацій з врахуванням вимог достовірності та безпеки даних під час змін у структурі мережі, що дозволяє здійснювати раціональний вибір параметрів системи.*

– *Удосконалено метод зберігання даних рекомендаційної системи, який відрізняється від відомих адаптацією до архітектури однорангових децентралізованих комп'ютерних мереж та використанням хеш-таблиць для*

зберігання даних про користувачів і об'єкти контенту та зв'язних списків для зберігання даних про рекомендації, що дозволило зменшити витрати часу і пам'яті на процеси обробки даних системи.

– *Удосконалено метод* пошуку та фільтрації даних рекомендаційною системою для формування рекомендацій користувачам, який відрізняється від відомих використанням запропонованого методу зберігання даних для збереження проміжних та підсумкових результатів обчислень, що дозволило використовувати його в однорангових децентралізованих комп'ютерних мережах та зменшити витрати часу і пам'яті при забезпеченні високої точності прогнозування вподобань користувачів.

Практична цінність роботи полягає у такому:

– Розроблена імітаційна модель процесів збору та обробки даних для рекомендаційної системи, що дозволила оцінити ймовірісно-технічні характеристики рекомендаційної системи децентралізованої однорангової комп'ютерної мережі. Отримано аналітичний вираз, за допомогою якого є можливість оцінити щільність розподілу ймовірностей часу ідентифікації стану вузлів децентралізованої рекомендаційної системи. Використання мультитригерного підходу та врахування вимог до достовірності та безпеки рекомендаційних повідомлень в порівнянні з відомими моделями дозволило підвищити точність результатів моделювання до 5%. Розроблено алгоритми моделювання рекомендаційної системи однорангової децентралізованої комп'ютерної мережі та процесів у ній, що дають можливість проводити тестування різних методів зберігання, пошуку та фільтрації даних у мережі.

– Розроблено алгоритми зберігання даних рекомендаційної системи однорангової децентралізованої комп'ютерної мережі, що дозволяють зменшити витрати пам'яті і часу на процеси зберігання та читання даних системи. Зокрема, розроблена система на основі запропонованих алгоритмів при використанні у децентралізованій одноранговій мережі показує наступні результати в порівнянні з найкращими результатами відомих систем – в 2,4 разів кращі результати по часу заповнення бази даних, в 1,7 разів кращі

результати по використаному об'єму пам'яті та в 2,5 разів кращі результати по часу генерації рекомендацій.

– Розроблено алгоритм пошуку та фільтрації даних рекомендаційною системою для формування рекомендацій користувачам в однорангових децентралізованих комп'ютерних мережах, що зменшує витрати часу і пам'яті при забезпеченні високої точності прогнозування вподобань користувачів. Точність розробленого алгоритму сягає до 0,84 в залежності від обраних параметрів системи.

Практичне значення отриманих результатів підтверджено відповідними актами впровадження. Результати дисертаційних досліджень впроваджені і використовуються у діяльності ІТ-компанії ТОВ "ОНІКС-СИСТЕМЗ", а також їх використано у навчальному процесі Центральноукраїнського національного технічного університету.

Особистий внесок здобувача. Усі наукові результати дисертаційної роботи автор отримав самостійно. У друкованих працях, опублікованих у співавторстві, здобувачеві належать: [22] – розробка математичної моделі збору та обробки даних для рекомендаційної системи однорангової децентралізованої комп'ютерної мережі для оцінки ймовірнісно-часових характеристик процесів формування та зміни рекомендацій; [23, 27, 34] – дослідження методів роботи та моделювання комп'ютерних мереж; [24, 33] – дослідження методів аналізу та прогнозування процесів в комп'ютерних мережах; [25, 32] – розробка способу формування рекомендацій користувачам однорангових децентралізованих комп'ютерних мереж з урахуванням кількості переходів до завантаження даних; [26] – розробка методу роботи рекомендаційної системи для однорангової децентралізованої комп'ютерної мережі; [28, 30, 37] – розробка методів зберігання даних рекомендаційної системи на основі різних структур даних у комп'ютерних мережах; [31] – дослідження принципів роботи децентралізованих структурованих комп'ютерних мереж; [29, 35, 36, 39] – дослідження методів зберігання даних

рекомендаційної системи у комп'ютерних мережах; [38] – дослідження методів зберігання даних у комп'ютерних системах за допомогою бінарних діаграм рішень; [40] – дослідження методів роботи рекомендаційних систем.

З робіт, що опубліковані у співавторстві, у дисертаційній роботі використовуються виключно результати, отримані особисто здобувачем.

Апробація результатів дисертації. Основні положення дисертаційної роботи доповідалися та обговорювалися на таких наукових конференціях та семінарах: IV та VI Міжнародних науково-практичних конференціях «Інформаційна безпека та комп'ютерні технології» (Україна, Кропивницький, 2021, 2023 рр.); Міжнародній науково-практичній Інтернет-конференції «Інновації та перспективні шляхи розвитку інформаційних технологій» (Україна, Черкаси, 2022 р.); 12-й Міжнародній науково-технічній конференції «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління» (Азербайджан–Україна–Словаччина, Баку–Харків–Жиліна, 2022 р.); Міжнародній науково-технічній конференції «Автоматика, комп'ютерно-інтегровані технології та проблеми енергоефективності в промисловості і сільському господарстві» (Україна, Кропивницький, 2022 р.); Міжнародній науково-практичній конференції «Цифрова трансформація суспільства» (Україна, Кропивницький, 2022 р.); XXIII Міжнародному науково-практичному семінарі «Комбінаторні конфігурації та їхні застосування» (Україна, Запоріжжя–Кропивницький, 2021 р.); 9-й Міжнародній науковій конференції «Інформація, комунікація, суспільство» (Україна, Львів, 2020 р.); II Міжнародній науково-практичній конференції «Інформаційна безпека та інформаційні технології» (Україна, Кропивницький, 2020 р.); Міжнародній науково-практичній конференції «Перспективні напрямки інформаційних і комп'ютерних систем та мереж, комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті» (Україна, Кропивницький, 2019 р.); XVII Міжнародному науково-

практичному семінарі «Комбінаторні конфігурації та їх застосування» (Україна, Кіровоград, 2015 р.).

Публікації. Основні положення дисертації опубліковано в 21 науковій праці, у тому числі: 10 наукових статей (з яких 3 входить до бази даних Scopus), 7 – у вітчизняних фахових наукових журналах), а також 11 матеріалів і тез доповідей на конференціях.

Структура роботи та її обсяг. Дисертація складається із анотації, вступу, чотирьох розділів, загальних висновків, списку використаної літератури після кожного розділу та додатків і містить 167 сторінок основного тексту, 21 рисуноків, 10 таблиць, 140 джерела у списку літератури та 12 сторінок додатків. Загальний обсяг роботи – 179 сторінок.

Список літератури до вступу

1. Narayanan M. Internet of Multimedia Things an Open Source P2P Publish/Subscribe Framework for Massive Semantic Content Spaces. *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, Greater Noida, India. 2023, P. 1632–1636. DOI: 10.1109/ICACITE57410.2023.10183252
2. Zhu J., Cheng W., Cui Y., Tang C., Dai Y., Li Y., Zeng L. Central Similarity Multi-View Hashing for Multimedia Retrieval. *Computer Vision and Pattern Recognition*. 2023. DOI: 10.48550/arXiv.2308.13774. URL: <https://arxiv.org/abs/2308.13774>
3. Liu W., Xie K., Pang L., Bailey J., Cao L., Zhang Y. Deep Learning for Search and Recommendation. *In Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM '22)*. Association for Computing Machinery, New York, NY, USA, 2022. P. 5171–5172. DOI: 10.1145/3511808.3557493
4. Ko H., Lee S., Park Y., Choi A. A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields.

- Electronics*. 2022, Vol. 11(1):141. DOI: <https://doi.org/10.3390/electronics11010141>
5. Badis L., Amad M., Aïssani D., Bedjguelal K., Benkerrou A. ROUTIL: P2P routing protocol based on interest links. *International Conference on Advanced Aspects of Software Engineering (ICAASE)*, Constantine, Algeria. 2016, P. 1–5. DOI: 10.1109/ICAASE.2016.7843852.
 6. Naeen H. M., Jalali M. A decentralized trust-aware collaborative filtering recommender system based on weighted items for social tagging systems”. *Social and Information Networks*. 2019. DOI: 10.48550/arXiv.1906.05143
 7. Thai P. D., Doan M., Liu W., Liu T., Li S., Zhou H. S., Dinh T. N. Blockchain Peer-to-Peer Network: Performance and Security. In: *Tran, D.A., Thai, M.T., Krishnamachari, B. (eds) Handbook on Blockchain. Springer Optimization and Its Applications*, Springer, Cham. 2022. Vol. 194. DOI: 10.1007/978-3-031-07535-3_2
 8. Forsbacka J., Sollenius G. A Peer-to-Peer Networking Framework for Scalable Massively Multiplayer Online Game Development in Unity. *Bachelor Thesis Master of Science in Computer Engineering + 300hp*. 2023. 40 p.
 9. Ali M., Asghar R., Ullah I., Ahmed A., Noor W., Baber J. Towards intelligent P2P IPTV overlay management through classification of peers. *Peer-to-Peer Netw*. 2022. Vol. 15. P. 827–838. DOI: 10.1007/s12083-021-01288-y
 10. Dias S., Sousa P. Implementation and Performance Analysis of Collaborative Peer-to-Peer Video-on-Demand Systems. *18th Iberian Conference on Information Systems and Technologies (CISTI)*, Aveiro, Portugal. 2023. P. 1–6, DOI: 10.23919/CISTI58278.2023.10211858.
 11. Benz J., Locher A. Peer-to-Peer Content Delivery Network. *Other thesis, OST Ostschweizer Fachhochschule*. 2023. URL: <https://eprints.ost.ch/id/eprint/1097>
 12. Ehiagwina F. O., Iromini N. A., Olatinwo I. S., Raheem K., Mustapha K. A State-of-the-Art Survey of Peer-to-Peer Networks: Research Directions, Applications and Challenges. *Management*. 2022. Vol. 14. P. 19–22.

13. Lua E. K., Crowcroft J., Pias M., Sharma R., Lim S. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. *IEEE Communications survey and tutorial*. 2004. URL: <https://snap.stanford.edu/class/cs224w-readings/lua04p2p.pdf>
14. Al-Asfoor M., Abed M.H. The Effect of the Topology Adaptation on Search Performance in Overlay Network. In: Jeena Jacob, I., Gonzalez-Longatt, F.M., Kolandapalayam Shanmugam, S., Izonin, I. (eds) *Expert Clouds and Applications. Lecture Notes in Networks and Systems*. Springer, Singapore. 2022. Vol. 209. URL: https://doi.org/10.1007/978-981-16-2126-0_7
15. Badis L., Amad M., Aïssani D., Abbar S. P2PCF: A collaborative filtering based recommender system for peer to peer social networks. *Journal of High Speed Networks*, P. 13-31. 2021. Vol. 27(1). DOI: 10.3233/jhs-210649
16. Ricci F., Rokach L., Shapira B., Kantor P.B. (Editors) *Recommender Systems Handbook*. Boston, Springer. 2011. 842 p. URL: 10.1007/978-0-387-85820-3
17. Burke R. Hybrid Web Recommender Systems. *The Adaptive Web. Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg. 2007. Vol. 4321. P. 377–408.
18. Mobasher B., Burke R., Bhaumik R., Williams C. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Transactions on Internet Technology*. 2007. Vol. 7(4). 41 p. DOI: 10.1145/1278366.1278372
19. Mobasher B., Burke R.D., Sandvig J.J. Model-based collaborative filtering as a defense against profile injection attacks. In *Proceedings of the 21st national conference on Artificial intelligence*. AAAI Press. 2006. Vol. 2. P. 1388–1393. URL: <https://dl.acm.org/doi/10.5555/1597348.1597409>
20. Williams A. C., Mobasher B., Burke R. Defending recommender systems: detection of profile injection attacks. *Service Oriented Computing and Applications*. 2007. P. 157–170.

21. Funk S. Netflix Update: Try This at Home. *The Evolution of Cybernetics – A Journal by Simon Funk*. 2006. URL: <https://sifter.org/~simon/journal/20061211.html>
22. Artemenko O., Kunanets O., Pasichnyk V. E-tourism recommender systems: a survey and development perspectives. *Econtechmod an international quarterly journal*. 2017. Vol. 6. No. 2. P. 91–95.
23. Пасічник В. В., Артеменко О. І. Особливості побудови просторово-орієнтованих мобільних туристичних рекомендаційних систем. *Матеріали XIV міжнародної конференції "Контроль і управління в складних системах (КУСС-2018)"*, м. Вінниця, ВНТУ, 15-17 жовтня 2018 р. С. 68–72.
24. Stekh Y., Artsibasov V. Adaptive clustering algorithm for recommender systems. *Вісник НУ "Львівська політехніка": Комп'ютерні системи проектування. Теорія і практика*, 2012. № 747. С. 75–78.
25. Stekh Y., Logvinenko A., Lobur M., Artsibasov V. Model and methods for building Web recommendation systems. *Proc. of the VI th International Conference on Computer Science and Information Technologies (CSIT 2011)*. Lviv. 2011. P. 314–316.
26. Лобур М. В., Стех Ю. В., Шварц М. Є. Побудова асоціативних правил для прогнозування рекомендацій в колаборативних рекомендаційних системах. *Квалілогія книги: Збірник наукових праць Української академії друкарства*. Львів, 2017. № 2 (32). С. 82–86.
27. Лобур М. В., Шварц М. Є., Стех Ю. В. Моделі і методи прогнозування рекомендацій для колаборативних рекомендаційних систем. *Вісник Національного університету «Львівська політехніка»: Інформаційні системи та мережі*. Львів. 2018. № 901. С. 68–75.
28. Vlasenko O., Veres O., Ilchuk P., Kots O., Levus Ya. Recommendation System for Leisure Time-Management in Quarantine Conditions. *Modern Machine Learning Technologies and Data Science Workshop*. 2022. Vol. 3312. P. 263–282. URL: <http://eprints.zu.edu.ua/36812/>

29. Верес О., Левус Я. Рекомендаційна система планування дозвілля в умовах карантину. *Вісник Національного університету «Львівська політехніка». Інформаційні системи та мережі*. 2022. Вип. 11. С. 127–144.
30. Bobadilla D., Lipizzi C. A Blockchain-Based Collaborative Filtering Recommendation System Based on Trust. *18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. 2021. DOI: 10.1109/iccwamtip53232.2021.9674128
31. Yoo H., Chung K. Mining-based lifecare recommendation using peer-to-peer dataset and adaptive decision feedback. *Peer-to-Peer Networking and Applications*. 2017. Vol. 11(6), P. 1309–1320. DOI: 10.1007/s12083-017-0620-2
32. Mikhav V., Semenov S., Meleshko Y., Yakymenko M., Shulika Y. Constructng the mathematical model of a recommender system for decentralized peer-to-peer computer networks. *Eastern-European Journal of Enterprise Technologies*. 2023. Vol. 4 (9(124)). P. 24–35. (ISSN 1729-3774). DOI: 10.15587/1729-4061.2023.286187. URL: <https://journals.urau.ua/eejet/article/view/286187> (SCOPUS)
33. Meleshko Ye., Drieieva H., Drieiev O., Yakymenko M., Mikhav V., Shymko S. (2023). A Method of Routing of Fractal-like Traffic with Prediction of Router Load for Reduce the Probability of Network Packet Loss. *Computational Linguistics and Intelligent Systems (COLINS 2023)*: CEUR-WS, Kharkiv, Ukraine / O. Cherednichenko, L. Chyrun, V. Vysotska (Ed.). 2023. Vol. 3403(III). P. 434–448. (ISSN 1613-0073). DOI: 10.5281/zenodo.8231189. URL: <https://ceur-ws.org/Vol-3403/paper34.pdf>. (SCOPUS)
34. Drieieva H., Drieiev O., Meleshko Ye., Yakymenko M., Mikhav V. A method of determining the fractal dimension of network traffic by its probabilistic properties and experimental research of the quality of this method. *Computational Linguistics and Intelligent Systems (COLINS 2022)*: CEUR-WS, Gliwice, Poland / V. Lytvyn (Ed.). 2022. Vol. 3154. P. 1694–1707.

(ISSN 1613-0073). DOI: 10.5281/zenodo.7892006. URL: <https://ceur-ws.org/Vol-3171/paper120.pdf>. (SCOPUS)

35. Мелешко Є. В., Дреєв О. М., Лавданський А. О. Модель рекомендаційної системи для комп'ютерних мереж типу P2P. *Вісник Черкаського державного технологічного університету*. 2023. Вип. 1. С. 52–60. DOI: 10.24025/2306-4412.1.2023.273495. URL: <http://vtn.chdtu.edu.ua/article/view/273495> (фахове видання категорії Б)

36. Міхав В. В., Мелешко Є. В. Метод роботи рекомендаційної системи у комп'ютерній мережі типу peer to peer. *Системи управління, навігації та зв'язку*. Полтава: ПНТУ, 2023. Т. 1(71). С. 112–117. DOI: 10.26906/SUNZ.2023.1.112. URL: <http://journals.nupp.edu.ua/sunz/article/view/2839> (фахове видання категорії Б)

37. Drieieva H., Meleshko Ye., Drieiev O., Mikhav V. Computer simulation model of a computer network with fractal traffic for testing routing algorithms. *Advanced Information Systems*. 2022. No. 6 (4). P. 11–18. DOI: 10.20998/2522-9052.2022.4.02. URL: <http://ais.khpi.edu.ua/article/view/268306>. (фахове видання категорії Б)

38. Міхав В. В., Мелешко Є. В., Якименко М. С., Шуліка Я. П. Розробка системи управління базою даних рекомендаційної системи для комп'ютерних та комп'ютерно-інтегрованих систем. *Центральноукраїнський науковий вісник. Технічні науки: зб. наук. пр.:* Кропивницький: ЦНТУ, 2022. Вип. 5 (36). С. 130–136. DOI: 10.32515/2664-262X.2022.5(36).2.130-136. URL: [http://mapiea.kntu.kr.ua/pdf/5\(36\)_II/19.pdf](http://mapiea.kntu.kr.ua/pdf/5(36)_II/19.pdf) (фахове видання категорії Б)

39. Міхав В. В., Мелешко Є. В., Якименко М. С., Бащенко Д. В. Методи зберігання даних рекомендаційної системи на основі зв'язних списків. *Системи управління, навігації та зв'язку*. Полтава: ПНТУ, 2021. Т. 4(66). С. 59–62. DOI: 10.26906/SUNZ.2021.4.059. URL: <http://journals.nupp.edu.ua/sunz/article/view/2422> (фахове видання категорії Б)

40. Міхав В. В., Мелешко Є. В., Якименко М. С. Метод зберігання даних рекомендаційної системи на основі бінарних діаграм рішень. *Системи управління, навігації та зв'язку*. Полтава: ПНТУ, 2020. Т. 2 (60). С. 85–89. DOI: 10.26906/SUNZ.2020.2.085. URL: <http://journals.nupp.edu.ua/sunz/article/view/1836> **(фахове видання категорії Б)**

41. Міхав В. В., Мелешко Є. В., Лавданський А. О. Дослідження принципів роботи однорангових децентралізованих структурованих комп'ютерних мереж. *Інформаційна безпека та комп'ютерні технології*: тези доп. VI Міжнар. наук.-практ. конф. (Кропивницький, 20–21 квіт. 2023 р.) / М-во освіти і науки України, Центральн. нац. техн. ун-т. Кропивницький: ЦНТУ, 2023. С. 51–52.

42. Міхав В. В., Мелешко Є. В., Дреєв О. М., Миронець І. В. Модель рекомендаційної системи для комп'ютерної мережі типу peer to peer. *Інновації та перспективні шляхи розвитку інформаційних технологій (ІПШРІТ-2022)*: тези доп. Міжнар. наук.-практ. Інтернет-конф., (Черкаси, 9 груд. 2022 р.) / упоряд.: Т. О. Прокопенко, Я. В. Тарасенко. М-во освіти і науки України, Черкас. держ. технол. ун-т, 2022. – С. 59–60.

43. Дреєва Г. М., Дреєв О. М., Мелешко Є. В., Якименко М. С., Міхав В. В. Дослідження якості роботи методу визначення фрактальної розмірності мережевого трафіку за його ймовірнісними властивостями. *Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління*: тези доп. 12-ї Міжнар. наук.-техн. конф., (Баку–Харків– Жиліна, 27–28 квіт. 2022 р.) / уклад. В. В. Косенко. Харків: ФОП Петров В. В., 2022. Т. 1. С. 126.

44. Дреєва Г. М., Мелешко Є. В., Міхав В. В. Програмна імітаційна модель комп'ютерної мережі для тестування алгоритмів маршрутизації трафіку. *Автоматика, комп'ютерно-інтегровані технології та проблеми енергоефективності в промисловості і сільському господарстві*: матеріали Міжнар. наук.-техн. конф., (Кропивницький, 10–11 листоп. 2022 р.) / М-во

освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: Ексклюзив-Систем, 2022. С. 44–45.

45. Міхав В. В., Мелешко Є. В., Якименко М. С., Башенко Д. В. Методи зберігання даних у рекомендаційних системах. *Цифрова трансформація суспільства*: тези доп. I Міжнар. наук.-практ. конф. (Кропивницький, 21–22 квіт. 2022 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2022. С. 44–45.

46. Міхав В. В., Мелешко Є. В., Якименко М. С. Дослідження можливостей графової бази даних Neo4j для застосування у розробці рекомендаційних систем соціальних мереж. *Комбінаторні конфігурації та їхні застосування*: тези доп. XXIII Міжнар. наук.-практ. семінару ім. А.Я. Петренюка, присвяч. 70-річчю ЛА НАУ (Запоріжжя–Кропивницький, 13–15 трав. 2021 р.) / за ред. Г. П. Донця. Кропивницький: Ексклюзив-Систем, 2021. С. 124–127.

47. Міхав В. В., Мелешко Є. В. Застосування методу delta-encoding для представлення даних рекомендаційної системи. *Інформаційна безпека та комп'ютерні технології*: тези доп. IV Міжнар. наук.-практ. конф. (Кропивницький, 15–16 квіт. 2021 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2021. С. 57.

48. Міхав В. В., Мелешко Є. В. Порівняння стратегій редагування бінарних діаграм рішень для роботи з графовими даними. *Інформація, комунікація, суспільство*: тези доп. 9-ої Міжнар. наук. конф. (Львів, 21–23 трав. 2020 р.) / М-во освіти і науки України, Нац. ун-т «Львівська політехніка». Львів: Вид-во Львів. політехніки, 2020. С. 17–18.

49. Міхав В. В., Мелешко Є. В. Метод оптимізації швидкодії бінарних діаграм рішень при представленні даних рекомендаційної системи. *Інформаційна безпека та інформаційні технології*: тези доп. II Міжнар. наук.-практ. конф. (Кропивницький, 2–3 квіт. 2020 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2020. С. 17.

50. Міхав В. В., Мелешко Є. В., Чабан О. О. Способи побудови рекомендаційних систем для соціальних мереж з врахуванням репутації користувачів. *Перспективні напрямки інформаційних і комп'ютерних систем та мереж, комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті*: тези доп. Міжнар. наук.-практ. конф. (Кропивницький, 13–14 лист. 2019 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2019. С. 86–87.

РОЗДІЛ 1

ДОСЛІДЖЕННЯ МЕТОДІВ РОБОТИ ОДНОРАНГОВИХ КОМП'ЮТЕРНИХ МЕРЕЖ ТА МОЖЛИВОСТІ РОЗРОБКИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ ДЛЯ НИХ

У цьому розділі проведено дослідження методів роботи однорангових комп'ютерних мереж, а саме: порівняльний аналіз методів пошуку даних у централізованих, децентралізованих неструктурованих та децентралізованих структурованих однорангових мережах [1-4]. Виявлено, що існуючі методи пошуку даних в однорангових комп'ютерних мережах можна покращити застосуванням рекомендаційних систем [1-3].

Проведено дослідження можливості розробки та застосування рекомендаційних систем у однорангових комп'ютерних мережах [1-3]. Розглянуто призначення, можливості та обмеження існуючих методів роботи децентралізованих рекомендаційних систем.

Також сформовано цілі та задачі дисертаційного дослідження.

1.1. Дослідження методів роботи однорангових комп'ютерних мереж

Однорангові комп'ютерні мережі засновані на принципі рівноправності вузлів мережі і характеризуються тим, що будь-які вузли можуть з'єднуватися між собою для надання послуг один одному на відміну від традиційної архітектури, у якій лише окрема категорія учасників (сервери) може надавати певні сервіси іншим [5-7].

Методи організації P2P-мереж можна розділити на наступні три види [5-15]:

- централізовані однорангові мережі, наприклад, BitTorrent [12];
- неструктуровані децентралізовані однорангові мережі; цей тип мереж використовує алгоритм пошуку Flooding, типовий представник – Gnutella [13];
- структуровані децентралізовані однорангові мережі, що найчастіше

будуються на основі розподілених хеш-таблиць, наприклад, на основі алгоритму Chord [14] або Kademlia [15].

Розглянемо відомі методи індексації та пошуку даних в P2P мережах. У різних типах однорангових мереж вони різні.

Централізована однорангова мережа. Інформація про індекс файлів зберігається на центральному сервері, кожен дочірній вузол повинен підключатися до нього, щоб знайти ресурси. Найбільшою перевагою такого підходу є просте обслуговування та швидке індексування. Однак, оскільки вся мережа значною мірою залежить від центрального сервера, якщо він вийде з ладу – вся система стане непрацездатною. Тож виникає проблема «єдиної точки відмови». Прикладом такої мережі є файлообмінний клієнт BitTorrent [10, 12]. Для роботи з ним потрібен «трекер» (tracker) – спеціалізований сервер, що працює за протоколом HTTP [12]. Трекер потрібен для того, щоб клієнти могли знайти один одного. Фактично, на трекері зберігаються IP-адреси, вхідні порти клієнтів та хеш-суми, що унікально ідентифікують об'єкти, які беруть участь у завантаженнях. За стандартом, імена файлів на трекері не зберігаються, і дізнатися їх по хеш-сумі не можна. Однак на практиці трекер часто, крім своєї основної функції, виконує і функцію невеликого веб-сервера [8]. Такий сервер зберігає файли метаданих і описи поширюваних файлів [12], надає статистику завантажень по різних файлах, показує поточну кількість підключених «пірів» (учасників файлообміну) тощо. Для підвищення працездатності системи у BitTorrent допускається існування декількох трекерів.

Неструктурована децентралізована однорангова мережа. Цей тип мережі використовує алгоритм пошуку Flooding і транслює запит файлу, який потрібно знайти, усім вузлам мережі для будь-якого пошуку [5, 8]. Коли вузол хоче завантажити файл, він створить запит із назвою файлу або ключовим словом і надішле його усім вузлам, підключеним до нього [5]. Якщо в цих вузлах є шуканий файл, встановлюється з'єднання з відповідним вузлом для завантаження. Якщо ні, запит продовжують пересилати на сусідні вузли, доки

не буде знайдено розташування файлу. Типовий представник цього методу для таких мереж – Gnutella рання версія. Слід зазначити, що коли масштаб мережі стає дуже великим, цей метод пошуку почне викликати «шторм трансляції», який серйозно знизить пропускну здатність мережі. Хоча цей метод дозволяє уникнути проблеми «єдиної точки відмови» централізованих однорангових мереж, він все одно дуже неефективний. Для усунення проблеми «шторму трансляції» було розроблено багато алгоритмів, але всі вони полягають у тому, що при пошуку файлу опитуються не всі вузли мережі, а тільки їх частина. Такі алгоритми знайдуть файл, який дійсно є в мережі не завжди, але вони є певним компромісом між швидкістю роботи мережі і ефективністю пошуку даних [8]. Найвідоміші з них подано нижче [5, 8, 10, 11].

- Метод широкого первинного пошуку (Breadth First Search, BFS) з додаванням до пошукового запиту параметра часу життя (Time-to-level, TTL). Параметр TTL визначає максимальну кількість переходів, якими можна пересилати запит. При типовому пошуку початкове значення для TTL становить зазвичай 5-7, яке зменшується щоразу, коли запит надсилається. Коли TTL дорівнює 0, повідомлення більше не передається.

- Метод випадкового широкого первинного пошуку (Random Breadth First Search, RBFS) було запропоновано як поліпшення "наївного" підходу BFS. У методі RBFS пошуковий запит пересилається лише частині вузлів мережі, вибраній у випадковому порядку. Якій саме частині вузлів – це параметр методу RBFS. Перевага RBFS полягає в тому, що не потрібна глобальна інформація про стан контенту мережі; вузол може отримувати локальні рішення так швидко, як це потрібно. З іншого боку, цей метод ймовірнісний. Тому деякі великі сегменти мережі можуть бути недосяжними.

- Інтелектуальний пошуковий механізм (Intelligent Search Mechanism, ISM). Поліпшення швидкості та ефективності пошуку інформації за допомогою цього методу досягається за рахунок мінімізації витрат на зв'язки, тобто на кількість повідомлень, що передаються між вузлами, та мінімізацію кількості вузлів, які опитуються для кожного пошукового запиту. Щоб

досягти цього, для кожного запиту оцінюються лише ті вузли, які найбільше відповідають даному запиту. Інтелектуальний пошуковий механізм складається з двох компонентів – профайлу (profile) та способу його ранжування, так званого рангу релевантності. Кожен вузол мережі буде інформаційний профайл для кожного із сусідніх вузлів. Профайл містить останні відповіді кожного із вузлів. За допомогою рангу релевантності здійснюється ранжування профайлів вузлів для вибору тих сусідніх, які надаватимуть найбільш релевантні документи на запит. Механізм профайлів використовується для того, щоб зберігати останні запити, а також кількісні характеристики результатів пошуку. При реалізації моделі ISM використовується єдиний стек запитів розміром $O(TN)$, в якому зберігається T запитів для N вузлів. Як тільки стек заповнюється, відбувається заміна "останнього найменш використовуваного" (Least Recently Used, LRU) для збереження останніх запитів. Функція "ранг релевантності" (Relevance Rank, RR) використовується вузлом P для виконання оперативної класифікації його сусідів та визначення тих, які слід опитувати першими за запитом q . Для обчислення рангу релевантності кожного вузла P , P_i порівнює q з усіма запитами q_j у структурі профайлу, для якого відомий список відповідей на попередні запити, та обчислюється $RR(P_i, q)$:

$$RR(P_i, q) = \sum_{j \leq Q} Sim(q_j, q)^a \cdot S(P_i, q_j), \quad (1.1)$$

де Q – множина запитів q_j , на які була відповідь у вузла P_i , $j \leq Q$; $S(P_i, q_j)$ – число результатів, що повертаються вузлом P_i на запит q_j ; метрика Sim розраховується за правилом косинуса:

$$Sim(q_j, q) = \frac{q_j \cdot q}{|q_j| \cdot |q|}, \quad (1.2)$$

і забезпечує вищий ранг вузла, який повертає більше результатів; параметр a – дозволяє збільшувати вагу запитів, найбільш подібних до поточного. Метод ISM ефективно працює у мережах, де вузли містять деякі спеціалізовані відомості. Зокрема, у мережі Gnutella якість пошуку дуже залежить від "оточення" вузла, з якого задається запит. Ще одна проблема методу ISM

полягає в тому, що пошукові повідомлення можуть зациклюватися, тому не в змозі досягти деяких частин мережі. Цю проблему можна вирішити, вибираючи невелику випадкову підмножину вузлів і додаючи до набору релевантних вузлів для кожного запиту, тоді механізм ISM став охоплювати більшу частину мережі.

– Методи "більшості результатів за минулою евристикою" (RES). В цих методах кожен вузол пересилає запит підмножині своїх вузлів, утвореній на підставі деякої узагальненої статистики. Запит у методі RES є задовільним, якщо видається Z або більше результатів (Z – деяка постійна). У методі RES вузол q пересилає запити до k вузлів, які видали найбільші результати останніх m запитів. Метод RES подібний до методу ISM, але використовує більш просту інформацію про вузли. Його головний недолік у порівнянні з ISM – відсутність аналізу параметрів вузлів, зміст яких пов'язаний із запитом. Тому метод RES характеризується швидше як кількісний, а не якісний підхід. RES хороший тим, що він маршрутизує запити у великі сегменти мережі (які можливо, також містять більш релевантні відповіді). Він також захоплює сусідів, які менш перевантажені, починаючи з тих, котрі зазвичай повертають більше результатів.

– Метод "випадкових блукань" (Random Walkers Algorithm, RWA) полягає в тому, що кожен вузол випадково пересилає повідомлення із запитом, що називається "посилкою", одному зі своїх вузлів. Щоб скоротити час, необхідний на отримання результатів, ідея однієї "посилки" розширена до " k -посилок" , де k – число незалежних посилок, послідовно запущених із пошукового вузла. Очікується, що " k -посилок" після T кроків досягнуть тих же результатів, що і одна посилка за kT кроків. Цей алгоритм нагадує метод RBFS, але за RBFS кожен вузол пересилає повідомлення запиту частині з його сусідів. До того ж, в RBFS передбачається експоненційне збільшення повідомлень, що пересилаються, а в методі випадкових блукань – лінійне. Обидва методи – і RBFS, і RWA – не використовують жодних явних

правил для того, щоб адресувати пошуковий запит до найбільш релевантного змісту.

– Метод "адаптивного ймовірнісного пошуку" (Adaptive Probabilistic Search, APS). В APS кожен вузол розгортає на своїх ресурсах локальний індекс, що містить значення умовних ймовірностей для кожного сусіда, який може бути обраний наступного переходу майбутнього запиту. Головна відмінність від RWA – те, що у APS вузол використовує зворотний зв'язок від попередніх пошуків (як умовних ймовірностей) замість цілком випадкових переходів. Тому метод APS часто дає кращі результати, ніж RWA.

Структурована децентралізована однорангова мережа. В даний час найбільшого поширення набула структурована децентралізована мережа [16-19], тобто мережа на основі DHT (distributed hash table, розподілена хеш-таблиця) [19]. DHT використовує більш структурований метод маршрутизації на основі пар ключ-значення. Для кожного вузла мережі обчислюється хеш-значення, воно стає його ідентифікатором. Для кожного файлу, що треба розмістити у мережі, також обчислюється хеш-значення такою ж хеш-функцією (воно стає ключем файлу), після чого файл записується на вузол, ідентифікатор якого максимально близький до ключа файлу за певною метрикою відстані. Кожен вузол мережі зберігає власну таблицю маршрутизації у мережі, де міститься інформація про вузли, ідентифікатори яких максимально близькі до нього за обраною метрикою. Прикладом практичної реалізації є мережа I2P, файлообмінний клієнт iMule, криптовалюта Ethereum [5-7, 20, 21]. Слід зазначити, що в реальних системах в тій чи іншій формі сервери у системі все ж присутні, і DHT часто відіграє лише допоміжну функцію, тобто маршрутизація в мережі побудована не тільки на ній. Зокрема, в Ethereum DHT використовується лише як ефективний механізм вибору партнерів. На даний момент існують наступні алгоритми роботи P2P мереж, що реалізують протокол DHT: Chord [14] і Kademlia [15]. І серед них найчастіше використовується алгоритм Kademlia. Розглянемо принцип роботи алгоритму Kademlia. Хеш значення найчастіше обчислюється

алгоритмом SHA-1, метрикою відстані є операція XOR. Таблиця маршрутизації зберігається у вигляді так званих K-bucket-тів. У кожному K-bucket-ті вузла зберігається інформація про K вузлів мережі, чия відстань до нього знаходиться в межах інтервалу $[2^i, 2^{(i+1)})$. Протокол Kademlia містить 4 типи повідомлень: 1) PING – необхідний для перевірки існування конкретного вузла у мережі; 2) STORE запит, що дає змогу розмістити інформацію на заданому вузлі; 3) FIND_VALUE запит, що дозволяє знайти значення за ключем; 4) FIND_NODE запит, що використовується для пошуку найближчих K вузлів до заданого ідентифікатора (поведінка подібна до FIND_VALUE, тільки ніколи не повертає значення, завжди вузли).

На даний час усі три види технологій побудови P2P мереж використовуються.

Як показує дослідження, у P2P мережах з різних причин може бути відсутній доступ до шуканих файлів. В такому разі користувачу доречно надати список рекомендацій на схожі файли. Але моделей та методів роботи рекомендаційних систем для P2P мереж в існуючих роботах знайдено не було. Дослідимо можливість побудови рекомендаційної системи для P2P мереж різного типу.

Розглянемо детальніше принципи роботи однорангових децентралізованих структурованих комп'ютерних мереж. Найчастіше для індексації та маршрутизації вони використовують розподілені хеш-таблиці (Distributed hash table, DHT). Для простоти будемо називати далі такі комп'ютерні мережі – P2P DHT мережі.

Узагальнені принципи роботи P2P DHT мереж наступні [5-11, 14, 15]:

1. *Ідентифікатори комп'ютерів.* Комп'ютери мають ідентифікатори $h \in H$, що змінюються у діапазоні $[0, 2^m-1]$. Вони визначається деякою хеш-функцією або залежить від черговості підключення комп'ютерів до мережі. Наприклад, перший комп'ютер у мережі отримує ідентифікатор $h_0=0$, i -тий комп'ютер одержує ідентифікатор $h_i=i$ або один із вільних ідентифікаторів, що вивільнилися внаслідок виходу з мережі деяких комп'ютерів, що раніше були

приєднані. Оскільки в децентралізованих мережах комп'ютери можуть приєднуватися та від'єднуватися від мережі, тобто учасники постійно змінюються, задача виявлення уже невикористовуваних індексів і надання їх новим комп'ютерам є важливою та по-різному вирішується у різних алгоритмах. Як правило, не тільки при приєднанні нового комп'ютера до мережі, а й з деякою періодичністю відбувається перевірка доступності вузлів та оновлення таблиць маршрутизації.

2. *Ідентифікатори файлів.* Файли також мають ідентифікатори $h \in H$, що змінюються у такому ж діапазоні, як і у комп'ютерів $[0, 2^m-1]$. Вони визначаються деякою хеш-функцією. Використовується так зване узгоджене хешування, що, на відміну від лінійного, передбачає зміну в середньому тільки K/n ключів, де K – число ключів, а n – число слотів. Популярним є застосування алгоритму хешування SHA-1 в P2P DHT мережах, в такому разі $m=160$, що дозволяє створити досить велику множину ідентифікаторів.

3. *Зберігання файлів.* Файли чи їх частини (при розподіленому зберіганні) або шляхи до них (при розподіленому індексуванні) зберігаються на комп'ютерах таким чином, щоб ідентифікатор комп'ютера та ідентифікатор файлу (його частини) співпадали. Або, при відсутності у мережі комп'ютера з потрібним ідентифікатором, вони повинні бути максимально близькі з наявних варіантів за деякою метрикою відстані. Надалі для простоти будемо розглядати розподілене зберігання файлів. Для забезпечення надійності зберігання інформації кожен файл дублюють на декілька комп'ютерів, для цього обирається q наявних у мережі комп'ютерів, ідентифікатори яких – найближчі за обраною метрикою відстані до ідентифікатора файлу. При зміні складу учасників мережі, за необхідності, відбувається перерозподіл частини файлів між комп'ютерами.

4. *Таблиця маршрутизації.* Зберігання таблиці маршрутизації також є розподіленим. Кожен комп'ютер містить інформацію, потрібну для доступу (мережеві адреси) найближчих N сусідів $X = \{x_{i-n}, \dots, x_{i+n}\}$ (рис. 1.1).

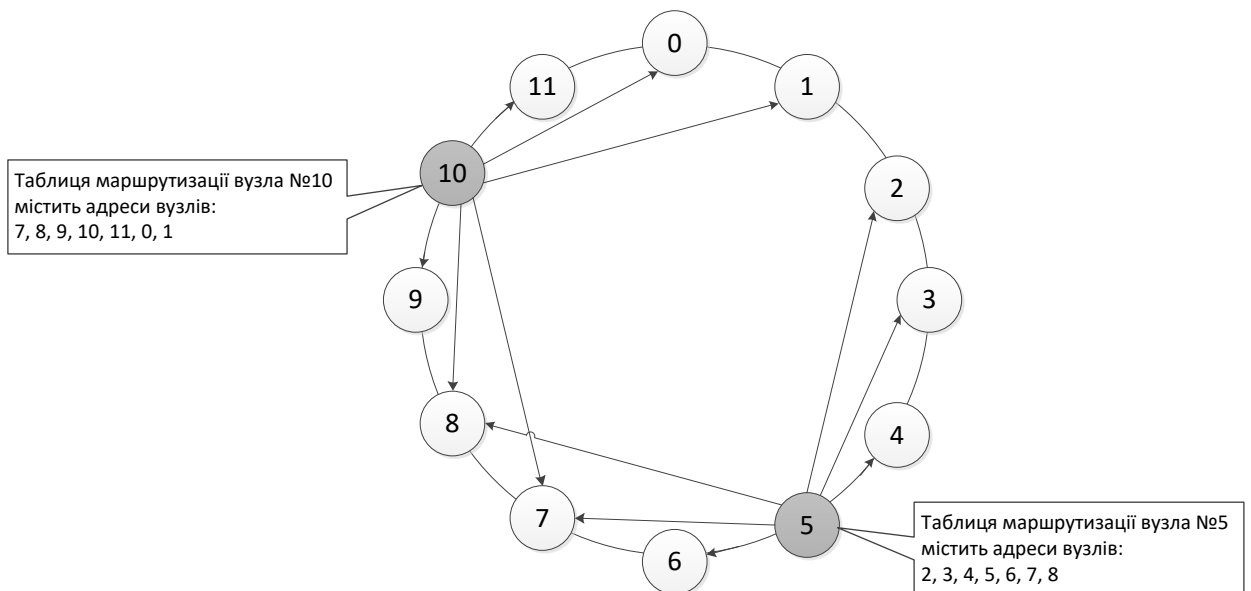


Рис. 1.1. Схематичне зображення принципу наповнення таблиць маршрутизації в однорангових децентралізованих структурованих комп'ютерних мережах: таблиці маршрутизації показані для комп'ютерів №5 та №10 при зберіганні 6 найближчих сусідів у мережі з 12 вузлів за принципами алгоритму Chord

Визначення множини X , а також розмір N цієї множини у кожному конкретному методі здійснюється по-різному. Наприклад, в алгоритмі Chord [14] таблиця маршрутизації комп'ютера містить $2r+1$ записів, тобто інформацію про сам комп'ютер, а також про r комп'ютерів, що мають найближчі менші номери серед наявних у мережі комп'ютерів, та про r комп'ютерів, що мають найближчі більші номери серед наявних комп'ютерів. Комп'ютери об'єднуються в кільце, тобто, останній комп'ютер посилається на перший, що викликає можливі проблеми з неправильним замиканням кільця, виникненням декількох кілець та розривом кільця при помилках в оновленні таблиць маршрутизації. В алгоритмі Kademlia [15] таблиця маршрутизації зберігається у вигляді так званих K-bucket-тів. У кожному K-bucket-ті вузла зберігається інформація про K вузлів мережі, відстань від яких до нього знаходиться в межах інтервалу $[2^i, 2^{(i+1)})$, як метрика відстані

використовується операція XOR. Якщо, наприклад, для створення ідентифікаторів застосовується хеш-функція SHA-1, то кількість K-bucket-тів на кожному вузлі буде 160. K – загальносистемне число, наприклад 20. Кожен K-bucket – це список, що містить не більше K -записів; тобто для мережі з $K=20$ кожен вузол матиме списки, що містять до 20 вузлів для певної відстані від себе. На практиці виходить, що кожен вузол зберігає інформацію про вузли, з якими будь-коли взаємодівав, протягом певного часу (наприклад, час життя запиту 24 години). Таким чином розмір таблиці маршрутизації змінюється динамічно, а деякі K-bucket-ти будуть порожніми в деякий момент часу.

5. *Пошук файлів.* Файли у мережі шукаються за їх хеш-значеннями. Комп'ютер, який здійснює пошук файлу, вибирає в своїй таблиці маршрутизації j комп'ютерів, ідентифікатори яких – найближчі за обраною метрикою відстані до ідентифікатора шуканого файлу. Кожен комп'ютер із множини вибраних $S = \{s_0, s_1, \dots, s_j\}$ перевіряє, чи міститься безпосередньо на ньому шуканий файл, і якщо ні, то пересилає пошукове повідомлення уже j комп'ютерам зі своєї таблиці маршрутизації, які найбільш підходять, і так – доки файл не буде знайдений. Чим більше значення s , тим менша ймовірність відмови та вищий рівень інформаційної безпеки, але й вище навантаження на мережу. Можна було б пересилати запит тільки на один комп'ютер з найближчим до шуканого ідентифікатором, але може виявитися, що він на даний час відсутній у мережі. Або ж цим комп'ютером володіє зловмисник, який перенаправить запит не на пошук та завантаження потрібного файлу, а на завантаження вірусу, в той же час при направленні запиту на декілька комп'ютерів можна буде обрати ту відповідь, яка прийде від більшості, що значно зменшить ймовірність інформаційної атаки.

6. *Основні запити до мережі.* Методи роботи P2P DHT мереж повинні містити також наступні функції:

- перевірка наявності сусідніх вузлів у мережі для оновлення таблиць маршрутизації;
- пошук вузлів за ідентифікатором;

– пошук файлів за ідентифікатором.

Зокрема, в протоколі алгоритму Kademlia для їх реалізації наявні наступні чотири типи запитів: 1) PING – необхідний для перевірки існування конкретного вузла у мережі; 2) STORE – дає змогу розмістити інформацію на заданому вузлі; 3) FIND_VALUE – дозволяє знайти значення за ключем; 4) FIND_NODE – використовується для пошуку найближчих K вузлів до заданого ідентифікатора (схожий на FIND_VALUE, тільки ніколи не повертає значення, завжди – вузли).

1.2. Дослідження можливості розробки рекомендаційних систем для однорангових комп'ютерних мереж

Найпростіше побудувати рекомендаційну систему у централізованих мережах – як ієрархічних, так і однорангових, оскільки у них наявний сервер, через який проходить інформація про завантаження файлів [22-25]. Отже, на основі статистики завантаження файлів різними користувачами можна побудувати рекомендаційну систему на колаборативній фільтрації. Її точність можна підвищити, якщо надати можливість користувачам оцінювати файли та зберігати оцінки на сервері.

Більш складною задачею є побудова рекомендаційної мережі для децентралізованих P2P мереж – як неструктурованих, так і структурованих. Оскільки сервери у таких системах відсутні, то кожен окремий комп'ютер буде самостійно розраховувати собі рекомендації на основі доступної йому інформації [26-28]. У децентралізованій мережі кожен комп'ютер системи може містити інформацію тільки про k найближчих (поняття та метрика визначення відстані залежить від конкретного алгоритму реалізації мережі) вузлів мережі, про розташовані на них файли та статистику завантажень (і, можливо, оцінки користувачів, надані файлам). Також він може запитати інформацію у своїх «сусідів» про їхніх «сусідів», про яких йому невідомо і т. ін.

В одноранговій мережі вузли – учасники мережі динамічно

під'єднуються та від'єднуються, при цьому можуть виникнути проблеми з якістю, достовірністю та безпекою даних інформаційної системи мережі [26-30], зокрема й рекомендаційної, адже нові вузли, що під'єднуються, можуть керуватися зловмисниками або бути уражені шкідливим програмним забезпеченням. Такі вузли можуть бути запрограмовані на викривлення або перехоплення даних рекомендаційної системи, оскільки інформаційні атаки на рекомендаційні системи є досить поширеним явищем [31-33].

Дослідження децентралізованих однорангових мереж показало, що питання створення рекомендаційних систем для такої архітектури є досить актуальним [26-30], але на практиці майже не розглядалося. Хоча в peer-to-peer мережах дуже гостро стоїть питання інформаційного пошуку [26-30], а рекомендаційні системи з різними алгоритмами фільтрації даних якраз могли б значно полегшити користувачам пошук корисного для них контенту та збільшили б інтерес до однорангових мереж для розподіленого зберігання контенту.

Було проведено огляд актуальних наукових досліджень [34-43] стосовно можливості розробки рекомендаційних систем для однорангових мереж.

У статті [34] наведено огляд актуальних досліджень у галузі рекомендаційних систем. Автори розглядають існуючі методи та моделі, що використовуються для побудови різних рекомендаційних систем. У статті наводяться різні методи фільтрації даних та формування рекомендацій, що використовуються в рекомендаційних системах. Однак ця стаття має аналітичний характер, у ній також не розглядається питання особливостей створення рекомендаційних систем для різних архітектур комп'ютерних мереж. У статті залишилось невирішеним питання оцінки ймовірнісно-технічних характеристик обробки даних.

У статті [35] пропонується децентралізована рекомендаційна система для соціальної мережі. Автори зазначають, що соціальні мережі містять багато цінної інформації, але зростання даних є однією з найбільших проблем, і це

привело до потреби в рекомендаційних системах. Для реалізації рекомендаційної системи в роботі використовується колаборативна фільтрація та метрика довіри поряд із метрикою подібності користувачів. Також пропонується децентралізований підхід, який розраховує схожість і довірчі відносини між користувачами розподіленим способом. Зазначається, що децентралізований підхід сприяє кращій продуктивності системи у створенні рекомендацій. Ця робота показує можливість розробки децентралізованих рекомендаційних систем, але тестування запропонованої моделі відбувається у централізованій ієрархічній комп'ютерній мережі. Тож запропонована модель рекомендаційної мережі, хоча і містить децентралізовані обчислення, не адаптована до архітектури та особливостей децентралізованих однорангових комп'ютерних мереж. Також в статті залишились невирішеними питання оцінки точності розробленої моделі та достовірності отриманих результатів.

В роботі [36] розглядаються проблема високої обчислювальної складності та проблеми представлення простору розподілених даних, які унеможливають колаборативну кластеризацію даних, яка, зокрема, використовується як важлива складова рекомендаційних систем в розподілених однорангових мережах. Автори пропонують нову серію алгоритмів колаборативної нечіткої кластеризації на основі випадкових ознак. В найбільш базовому алгоритмі, кожен вузол в розподіленій P2P-мережі спочатку відображає свої дані в низькорозмірний простір випадкових ознак з апроксимацією заданого ядра за допомогою методу відображення випадкових Фур'є-ознак. Потім кожен вузол незалежно шукає кластери зі своїми локальними даними та колаборативними знаннями від своїх сусідніх вузлів. І розподілена кластеризація виконується серед усіх вузлів мережі до досягнення глобального консенсусного результату, тобто всі вузли мають однакові центри кластерів. Відкритими питаннями залишаються наступні: вибір оптимальних ядер для кластеризації, залежність від випадкових знак, необхідність досягнення глобального консенсусу. Також у статті не наведено моделі

рекомендаційної системи для децентралізованої мережі, хоча запропоновані алгоритми колаборативної кластеризації будуть корисними при її розробці.

В роботі [37] пропонується алгоритм побудови розподіленої рекомендаційної системи на основі колаборативної фільтрації та розподілених хеш-таблицях DHT, що дозволяє знаходити найближчих «сусідів» для користувачів, яким обчислюються рекомендації у децентралізованій мережі. Алгоритм шукає інформацію про «сусідів» користувача відповідно до «fuzzy critical value», що генерується за «extreme score» користувача. Робота зосереджена на знаходженні найбільш відповідних «сусідів» для користувача, щоб якісно реалізувати колаборативну фільтрацію у децентралізованій мережі, проте не розглядаються інші проблеми розробки розподілених децентралізованих рекомендаційних систем, зокрема їх достовірність та інформаційна безпека.

В роботі [38] пропонується метод рекомендації довіреного вузла для міждоменної колаборативної фільтрації на основі SDN. За допомогою методу міждоменної колаборативної фільтрації шукаються довірені вузли мережі (мобільні пристрої), які демонструють достовірність на основі своїх дій. Експерименти з моделюванням підтверджують, що під час вибору надійного вузла метод, запропонований у цій статті, має високу точність рекомендацій і невеликий час роботи. Ця робота загалом може бути корисною для розробки децентралізованих рекомендаційних систем. Але вона зосереджена на конкретній сфері застосування рекомендаційних систем і вирішує вузьку специфічну задачу, тож може не підходити до вирішення інших задач щодо використання рекомендаційних систем. Крім того, в статті залишились невирішеними питання оцінки точності розробленої моделі з урахуванням вимог до достовірності та безпеки.

У статті [39] пропонується підхід до розробки децентралізованих рекомендаційних систем, заснований на використанні технології блокчейн та смарт-контрактах. Запропонована система побудована на основі технології Distributed Ledger, яка працює без будь-яких централізованих повноважень

і підтримує як децентралізовані рейтинги, так і ранжування різних елементів. Проведення експериментів із залученням запропонованої системи відбувалося на платформі Ethereum та продемонструвало доцільність запропонованого підходу з точки зору продуктивності та вартості. Однак, як і у статті [23], автори не розглядають деякі можливі обмеження застосування технології блокчейн у рекомендаційних системах, такі як висока складність алгоритмів, тривалий час обчислень та недостатня універсальність запропонованого підходу.

В роботі [40] пропонується децентралізована рекомендаційна система для однорангової мережі на основі популярної моделі колаборативної фільтрації та блокчейн технологій. Вона застосовується для створення системи розумних контрактів для підприємств і підвищення інформаційної безпеки їх онлайн-активностей. Автори пропонують новий алгоритм, який дозволяє вирішувати проблему централізованого зберігання та обробки даних у рекомендаційних системах. Описується лише один конкретний метод побудови децентралізованих рекомендаційних систем на основі блокчейну. Це знижує практичну цінність даної роботи для ряду специфічних рекомендаційних систем, в яких технологія блокчейн не є ефективною. Прикладом таких систем є ті, що потребують високої швидкості обробки даних та швидкого прийняття рішень, системи з високим ступенем конфіденційності даних тощо.

В роботі [41] запропонована рекомендаційна система P2PCF. Автори використовують підхід колаборативної фільтрації для рекомендації контенту в соціальних мережах P2P. Як стверджують автори статті, P2PCF забезпечує збереження конфіденційності. В той же час, запропонований підхід передбачає, що матриця рейтингів розподілена між користувачами таким чином, що кожен користувач бачить тільки взаємодії, зроблені його друзями на його часовій шкалі. Це значно зменшує достовірність вхідних даних, та знижує точність результатів моделювання.

В роботі [42] автори, навпаки, більшою мірою приділяють увагу

достовірності отриманих даних, збільшуючи можливі інформаційні джерела та розглядаючи можливість їх сумісного використання. Але питання безпеки в запропонованій моделі не розглядається. Крім того, технічні та практичні можливості використання усього спектру запропонованих джерел рекомендаційних даних в статті не обґрунтовано. З цього можна зробити висновок про необхідність фільтрування та узагальнення вхідних даних, моделювання та оптимізації за критерієм максимальної точності з урахуванням вимог безпеки та достовірності.

В роботі [43] запропоновано метод розрахунку довіри до P2P на основі аналізу мережевої поведінки. Метод спирається на основні елементи соціальної психології щодо встановлення інтимних стосунків між людьми, враховує вплив поведінки вузлів на довіру та використовує сукупну довіру, поточну довіру, довіру зворотного зв'язку та сукупну довіру до зловживань для розрахунку та оновлення значення довіри, щоб покращити динамічну адаптивність та відмовостійкість однорангових мереж. В майбутньому розглянуті в статті елементи можуть бути використані в моделі довіри до рекомендацій. Але в статті не наведено, яким чином можна об'єднати більшість елементів у єдиній моделі рекомендаційної системи. І тому ці матеріали потребують доопрацювання в площині адаптації окремих наведених елементів довіри до вимог безпеки та достовірності рекомендаційних даних в узагальненій моделі рекомендаційної системи для децентралізованих однорангових комп'ютерних мереж.

Можливо, питання точності результатів моделювання однорангових рекомендаційних систем залишилися не вирішеними тому, що використовуваний математичний апарат не завжди надає можливість знаходити ефективні рішення.

При вирішенні завдань підвищення точності результатів дослідники вимушено стискаються з проблемами внесення додаткових вхідних даних та ускладнення моделей. В цьому випадку корисним може бути досвід використання технології GERT-мережевого моделювання.

GERT використовує графічну нотацію, що дозволяє легко представити послідовність та залежності подій в проекті. Це полегшує моделювання та аналіз проекту, зокрема визначення ймовірнісних та часових характеристик.

GERT-мережева технологія дає змогу враховувати невизначеність імовірнісних величин, таких як час виконання подій або затримки, за допомогою випадкових величин. Це дозволяє більш реалістично моделювати різні сценарії та враховувати можливі ризики та невизначеності у результатах.

Крім того, GERT-мережеве моделювання дозволяє враховувати одночасні або паралельні події, які можуть відбуватися одночасно. Це важливо для точного визначення часу виконання процесів.

Також для різних архітектур комп'ютерних мереж рекомендаційні системи треба розробляти по-різному, а існуючі рекомендаційні системи орієнтовані на централізовані ієрархічні мережі, і практично відсутні математичні моделі їх роботи для децентралізованих однорангових мереж.

1.3. Дослідження методів та структур даних для зберігання інформації рекомендаційної системи в однорангових комп'ютерних мережах

У наш час існує багато різних систем управління базами даних; окрім реляційних баз даних широке застосування отримують бази даних типу NoSQL [44, 45]. Саме бази даних типу NoSQL якнайкраще можуть підійти для зберігання даних в однорангових комп'ютерних мережах.

В однорангових децентралізованих комп'ютерних мережах доцільно використовувати СУБД саме NoSQL типу, тому що це дозволить враховувати особливості архітектури мережі та адаптувати розроблюване програмне забезпечення до неї.

СУБД типу NoSQL можуть бути реалізовані різними методами, зокрема, як Сховища типу «ключ-значення» (Key-value stores), Масштабовані розподілені сховища (Column Family (Bigtable) stores), графові СУБД (Graph

Stores), документо-орієнтовані СУБД (Document Stores) тощо [44-51].

Розглянемо детальніше основні моделі збереження інформації у системах управління базами даних [44-51]:

1. *Прості моделі даних.* Дані зберігаються у звичайних файлах. Наприклад, у електронних таблицях – у файлах формату .csv або інших. Підходять для зберігання даних з простою структурою.

2. *Ієрархічні моделі даних.* Можуть бути представлені файловою системою, за допомогою якої можна здійснити деревовидне структурування даних шляхом розміщення їх у каталогах та файлах. Кожен запис може мати не більше одного батьківського запису і не можливо реалізувати відношення «багатьох-до-багатьох». Прикладом таких баз даних можуть бути різні файлові системи.

3. *Реляційні моделі даних.* Найпопулярніші моделі загального призначення. Дані представлені у вигляді таблиць. Окремий запис представлений у вигляді рядка таблиці. Для доступу до даних використовується структурована мова запитів SQL. Ці моделі мають ряд недоліків, що призводять до надмірності даних, які можуть приводити до різних аномалій та порушень цілісності даних. Для усунення цих недоліків структури використовується нормалізація. Приклади – MySQL, SQLite, Visual FoxPro, Access, Oracle.

4. *Модель даних «ключ-значення».* Належать до NoSQL типу. Для зберігання даних використовуються хеш-таблиці, які зберігають колекцію записів, що містять множину різних полів даних. Записи зберігаються та вилучаються з використанням ключа. Немає жорсткої схеми відношень між даними. Хеш-таблиці забезпечують швидкий доступ до даних, але ресурсомісткі. Найпопулярнішим прикладом є БД Redis.

5. *Документні моделі даних.* Належать до NoSQL типу. В основі лежать документні сховища, що мають структуру дерева. Дані зберігаються в структурованих форматах JSON, BSON або XML. Кожен документ може мати свою внутрішню структуру. Запити до бази даних дозволяють знайти

документ або частину документу. Документні моделі даних застосовуються в системах управління змістом, видавничій справі, документальному пошуку. Приклади – CouchDB, MongoDB, RethinkDB, eXist.

6. *Графові моделі даних.* Належать до NoSQL типу. Дані зберігаються у вигляді вершин та ребер графів. Вершини та ребра можуть мати будь-яку кількість пов'язаних з ними властивостей. Часто застосовують для моделювання соціальних графів, семантичної павутини тощо. Можуть бути високопродуктивними, є більш наглядними та простими для внесення змін. Приклади – Neo4j, JanusGraph, JanusGraph, ArangoDB.

7. *Комбіновані моделі даних.* Об'єднують переваги SQL та NoSQL підходів у побудові баз даних. Мають можливість горизонтального масштабування, велику продуктивність, гнучкість, але й велику ресурсомісткість і необхідність спеціалізованих знань для роботи з ними. Приклади – MemSQL, VoltDB, Calvin, Calvin.

Вибір моделі збереження даних залежить від властивостей даних, що потребують обробки (зокрема, складності їх структури), сфери застосування і апаратних можливостей.

Все частіше для зберігання даних рекомендаційних систем та інших додатків починають використовувати графові моделі [52-57]. І це відбувається з огляду на ряд переваг графових моделей, зокрема зручність представлення даних для програміста та можливість легкого масштабування і створення мережеских баз даних. Яскравим прикладом такого підходу є побудова рекомендаційних систем із застосуванням графової СУБД Neo4j [52]. Графові моделі СУБД надають не лише зручний формат зберігання даних, а й зручний формат запитів. В документації до Neo4j є приклади реалізації алгоритмів формування рекомендацій запитом до цієї СУБД, що ілюструє її придатність для використання в рекомендаційних системах.

Спосіб зберігання даних рекомендаційної системи є важливим з точки зору якості її роботи, швидкості, можливостей масштабування, зручності

виконання основних операцій з даними для формування рекомендацій.

Зберігати графові структури у пам'яті комп'ютерів можна різними методами. Наявність такої великої кількості різних методів реалізації баз даних та представлення інформації, що можна використати при побудові рекомендаційних систем, викликає необхідність порівняльного аналізу та вибору оптимального методу і структури даних для зберігання інформації рекомендаційних систем. Було проведено дослідження різних структур даних, які можна використати для створення бази даних рекомендаційної системи [58-62], зокрема – наведених нижче.

Бінарні діаграми рішень (BDD) – економна форма представлення булевих функцій у вигляді орієнтованого ациклічного графу [63]. Вершини графу представляють аргументи функції, листки – її двійкові значення. Для додавання і вилучення ребер та зміни ваги ребер необхідно мати можливість редагувати дані графу. БДР дають можливість зберігати дані у стисненому вигляді та швидко отримувати значення функції за її параметрами, але редагування БДР вимагає складних обчислень [63]. При представленні булевих функцій у формі БДР стало можливим розв'язувати багато проблем, які при традиційних представленнях структур нерозв'язні через значну розмірність таких представлень і складність операцій над ними. БДР можуть успішно застосовуватися фактично в кожній галузі, де потрібно обробляти дискретні структури даних.

Хеш-таблиця (hash map) – структура даних, у якій пошук елементу здійснюється на основі його ключа. На розташування елемента у хеш-таблиці вказує хеш-значення його ключа. Якщо декілька елементів мають однаковий хеш, то виникає колізія. Існує два методи розв'язання колізій – закрита та відкрита адресації. При закритій адресації кожен елемент таблиці – це зв'язний список, і усі елементи з однаковим хешем додаються до одного списку. Це є найпростішим способом розв'язання колізій, але потребує використання додаткової пам'яті для вказівників і не дозволяє використовувати переваги

кешування при обході елементів хеш-таблиці. При відкритій адресації у випадку колізії обирається нова позиція елементу. Нова позиція може обиратися як за допомогою додаткової хеш-функції, так і шляхом зміщення позиції на декілька елементів. Пошук елементу повторюється, доки не буде досягнуто порожнього запису у таблиці. Відкрита адресація використовує фіксований об'єм пам'яті і не потребує додаткових вказівників, але для ефективності операцій вставки і пошуку таблиця має бути заповнена не більш ніж на 50%, тож це спричиняє додаткові витрати пам'яті.

В-дерево (b-tree) – структура даних представлена збалансованим та сильно розгалуженим деревом пошуку. Кожен вузол В-дерева, крім листків, є упорядкованим списком, у якому чергуються ключі і вказівники на нащадків. Ключі вузла вказують інтервал, у якому знаходяться ключі нащадка. **В+-дерево (B+-tree)** відрізняється тим, що воно зберігає усі значення у листкових вузлах, а листкові вузли мають посилання на сусіда, завдяки чому можна обійти усі значення без обходу всього дерева. Завдяки великій розгалуженості дерева підтримується мала висота дерева, що дозволяє переглядати невеликий об'єм даних за один прохід, а завдяки правилам побудови значення зберігаються у порядку зростання ключа.

Зв'язний список (linked list) – структура даних, у якій кожен елемент містить поля даних та вказівник на наступний елемент. Основна перевага цієї структури полягає у сталому часі додавання нового елементу. Проте для кожного елементу потрібно виділяти новий блок пам'яті, тому менеджер пам'яті спричиняє значні затримки і накладні витрати пам'яті в процесі роботи.

Розгорнутий зв'язний список (unrolled linked list) – зв'язний список, кожен елемент якого містить масив логічних елементів. Це дозволяє об'єднати переваги масивів та зв'язних списків. Об'єднання блоків логічних елементів у список дозволяє додавати нові елементи без зміни розміру блоку пам'яті, економити пам'ять на вказівниках та ефективніше використовувати кеш процесора завдяки послідовному розташуванню елементів. При послідовному

заповненні списку гарантується, що незаповненим лишиться не більше одного блоку елементів.

У розділі 3.4 цієї роботи проведено експерименти для порівняльного аналізу різних методів зберігання інформації рекомендаційної системи децентралізованої однорангової комп'ютерної мережі на основі вказаних вище структур даних.

1.4. Цілі та задачі дисертаційного дослідження

У цій дисертаційній роботі вирішується науково-практична задача, що полягає у підвищенні ефективності методів обробки даних в однорангових децентралізованих комп'ютерних мережах шляхом розробки і удосконалення моделей і методів роботи рекомендаційних систем для цієї архітектури мереж.

Як показали проведені дослідження, P2P комп'ютерні мережі знову актуальні й широко використовуються у наш час завдяки численним перевагам, які вони пропонують. Якщо спочатку вони набули популярності завдяки файлообмінним сервісам, то зараз їх використовують й у багатьох інших сферах, зокрема для реалізації криптовалют і додатків для управління цифровими активами, інтернет-телефачення, стрімінгу, онлайн комп'ютерних ігор, децентралізованих соціальних мереж, Інтернету речей тощо (рис. 1.2).

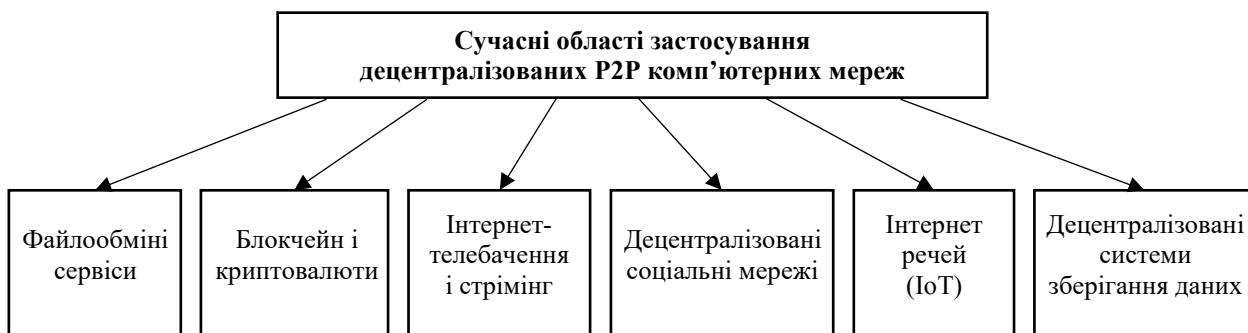


Рис. 1.2. Сучасні області застосування децентралізованих P2P комп'ютерних мереж

В однорангових децентралізованих мережах існує кілька варіантів доступу до контенту, який вони містять (рис. 1.3):

- через прямі посилання, розміщені на веб-ресурсах;
- через пошукові системи однорангової мережі;
- через рекомендаційні системи однорангової мережі.



Рис. 1.3. Способи доступу до контенту у P2P комп'ютерних мережах

В однорангових мережах задача пошуку даних значно ускладнюється порівняно зі звичайними мережами через особливості архітектури таких мереж. Рекомендаційні системи дають змогу полегшити пошук при великій кількості контенту, доповнюючи або заміняючи класичну пошукову видачу рекомендаціями. В P2P мережах їх застосування може мати додаткову користь. Через проблеми індексації і пошуку раніше додані файли можуть бути недоступні для користувача P2P мережі. Якщо користувач не може знайти шуканий файл, можна надати йому список рекомендацій з врахуванням його вподобань і пошукового запиту.

Таким чином, забезпечення якісної обробки даних, зокрема їх пошуку та фільтрації, є однією з важливих задач, які мають вирішуватися при забезпеченні якості та ефективності роботи однорангових децентралізованих комп'ютерних мереж. Одним із перспективних та недостатньо досліджених методів вирішення цієї задачі є розробка моделей та методів роботи рекомендаційних систем для однорангових децентралізованих комп'ютерних мереж.

Для вирішення поставленої науково-практичної задачі була сформульована наступна мета дисертаційної роботи – зменшення витрат часу

та пам'яті на роботу рекомендаційної системи в однорангових децентралізованих комп'ютерних мережах при достатній точності створення рекомендацій користувачам.

Мета дисертаційної роботи визначає необхідність розв'язання таких основних задач:

1. Розробити математичну модель збору і обробки даних для рекомендаційної системи однорангової децентралізованої комп'ютерної мережі.

2. Удосконалити метод зберігання даних рекомендаційної системи з врахуванням особливостей архітектури однорангових децентралізованих комп'ютерних мереж.

3. Удосконалити метод пошуку та фільтрації даних рекомендаційною системою для формування рекомендацій з врахуванням особливостей архітектури однорангових децентралізованих комп'ютерних мереж.

4. Провести експериментальне дослідження запропонованих методів роботи рекомендаційних систем для однорангових децентралізованих комп'ютерних мереж.

Висновки до розділу 1

У цьому розділі проведено дослідження методів роботи однорангових комп'ютерних мереж. Здійснено порівняльний аналіз методів пошуку даних у централізованих, децентралізованих неструктурованих та децентралізованих структурованих однорангових мережах. Виявлено, що існуючі методи пошуку даних в однорангових комп'ютерних мережах можна покращити застосуванням рекомендаційних систем.

Також проведено дослідження можливості розробки та застосування рекомендаційних систем у однорангових комп'ютерних мережах. Розглянуто призначення, можливості та обмеження існуючих методів роботи децентралізованих рекомендаційних систем.

У першому розділі також сформовано цілі та задачі дисертаційного дослідження. Для вирішення поставленої науково-практичної задачі була сформульована наступна мета дисертаційної роботи – зменшення витрат часу та пам'яті на роботу рекомендаційної системи в однорангових децентралізованих комп'ютерних мережах при достатній точності створення рекомендацій користувачам.

Список літератури до розділу 1

1. Мелешко Є. В., Дреєв О. М., Лавданський А. О. Модель рекомендаційної системи для комп'ютерних мереж типу P2P. *Вісник Черкаського державного технологічного університету*. 2023. Вип. 1. С. 52–60. DOI: 10.24025/2306-4412.1.2023.273495. URL: <http://vtn.chdtu.edu.ua/article/view/273495> **(фахове видання категорії Б)**

2. Міхав В. В., Мелешко Є. В. Метод роботи рекомендаційної системи у комп'ютерній мережі типу peer to peer. *Системи управління, навігації та зв'язку*. Полтава: ПНТУ, 2023. Т. 1(71). С. 112–117. DOI: 10.26906/SUNZ.2023.1.112. URL: <http://journals.nupp.edu.ua/sunz/article/view/2839> **(фахове видання категорії Б)**

3. Міхав В. В., Мелешко Є. В., Дреєв О. М., Миронець І. В. Модель рекомендаційної системи для комп'ютерної мережі типу peer to peer. *Інновації та перспективні шляхи розвитку інформаційних технологій (ІПШРІТ-2022): тези доп. Міжнар. наук.-практ. Інтернет-конф., (Черкаси, 9 груд. 2022 р.)* / упоряд.: Т. О. Прокопенко, Я. В. Тарасенко. М-во освіти і науки України, Черкас. держ. технол. ун-т, 2022. С. 59–60.

4. Міхав В. В., Мелешко Є. В., Лавданський А. О. Дослідження принципів роботи однорангових децентралізованих структурованих комп'ютерних мереж. *Інформаційна безпека та комп'ютерні технології: тези доп. VI Міжнар. наук.-практ. конф. (Кропивницький, 20–21 квіт. 2023 р.)* / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2023. С. 51–52.

5. Riposo Ju. Diffusion on the Peer-to-Peer Network. *LAP LAMBERT Academic Publishing*. 2022. 100 p.
6. Koo S. G. M. Multimedia Content Distribution Using Peer-to-Peer Overlay Networks: The Design and Analysis of the Next Generation Peer-to-Peer Networks. *VDM Verlag Dr. Müller*. 2008. 88 p.
7. Milojevic D.S., Kalogeraki V., Lukose R., Nagaraja K., Pruyne J., Richard B., Rollins S., Xu Z. Peer-to-peer computing. *Technical Report HPL-2002-57, HP Labs*. 2002. 51 p. URL: <https://www.cs.kau.se/cs/education/courses/dvad02/p2/seminar4/Papers/HPL-2002-57R1.pdf>
8. Zeinalipour-Yazti D., Kalogeraki V., Gunopulos D. Information retrieval techniques for peer-to-peer networks. *Computing in Science & Engineering*, 2004. Vol. 6. No. 4. P. 20–26. DOI: 10.1109/MCSE.2004.12
9. Lua E. K., Crowcroft J., Pias M., Sharma R., Lim S. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. *IEEE Communications survey and tutorial*. 2004. URL: <https://snap.stanford.edu/class/cs224w-readings/lua04p2p.pdf>
10. Kalogeraki V., Gunopulos D., Zeinalipour-Yazti D. A Local Search Mechanism for Peer-to-Peer Networks. *Proc. of CIKM'02*, McLean VA, USA, 2002. URL: <http://alumni.cs.ucr.edu/~csyiazti/downloads/papers/cikm02/cikm02.pdf>
11. Zeinalipour-Yazti D. Information Retrieval in Peer-to-Peer Systems. *M.Sc Thesis, Dept. of Computer Science, University of California Riverside*. 2003. URL: <http://alumni.cs.ucr.edu/~csyiazti/papers/msc/html/>
12. The BitTorrent Protocol Specification. 2017. URL: http://www.bittorrent.org/beps/bep_0003.html
13. Gnutella Protocol Development. 2003. URL: <https://rfc-gnutella.sourceforge.net>
14. Stoica I., Morris R., Karger D.R., Kaashoek M.F., Balakrishnan H. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. *ACM SIGCOMM Computer Communication Review*. 2001. Vol. 31(4). DOI:

15. Kademlia: A Design Specification. 2010. URL: <https://xlattice.sourceforge.net/components/protocol/kademlia/specs.html>
16. Shukla N., Datta D., Pandey M. Towards software defined low maintenance structured peer-to-peer overlays. *Peer-to-Peer Netw Appl.* 2021. Vol. 14. P. 1242–1260. DOI: 10.1007/s12083-021-01112-7
17. Klots Y., Muliar I., Cheshun V., Burdyug O. Use of distributed hash tables to provide access to cloud services. *Collection of scientific works of the Military Institute of Kyiv National Taras Shevchenko University*, 2020. Vol. 11(67). P. 85–95.
18. Shen Yb., Gadekallu T. R. Resource Search Method of Mobile Intelligent Education System Based on Distributed Hash Table. *Mobile Netw Appl.* 2022. Vol. 27. P. 1199–1208. DOI: 10.1007/s11036-022-01940-8
19. Ratnasamy S., Stoica I., Shenker S. Routing Algorithms for DHTs: Some Open Questions. Peer-to-Peer Systems. *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg. 2002. Vol. 2429. P. 45–52. DOI: 10.1007/3-540-45748-8_4
20. Bellet A., Guerraoui R., Taziki M., Tommasi M. Personalized and Private Peer-to-Peer Machine Learning. *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*. 2018. Vol. 84. P. 473–481. URL: <http://proceedings.mlr.press/v84/bellet18a/bellet18a.pdf>
21. Ricardo D. Introduction to Bitcoin: Understanding Peer-to-Peer Networks, Digital Signatures, the Blockchain, Proof-of-Work, Mining, Network Attacks, Bitcoin Core ... Safety. *Expiscor Books*; 1st ed. 2021. 60 p.
22. Fayyaz Z., Ebrahimian M., Nawara D., Ibrahim A., Kashef R. Recommendation systems: Algorithms, challenges, metrics, and business opportunities. *Applied Sciences*. 2020. Vol. 10(21), 7748. DOI: 10.3390/app10217748 URL: <https://www.mdpi.com/2076-3417/10/21/7748>
23. Falk K. Practical recommender systems. Simon and Schuster. 2019. 432 p.

24. Ko H., Lee S., Park Y., Choi A. A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields. *Electronics*. 2021. Vol. 11(1), 141. DOI: 10.3390/electronics11010141
25. Mohammed A. S., Meleshko Y., Balaji S. B., Semenov S., Collaborative Filtering Method with the use of Production Rules. *International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, Dubai, United Arab Emirates. 2019. P. 387–391, DOI: 10.1109/ICCIKE47802.2019.9004257
26. Fulber-Garcia V. Centralized Computing vs. Distributed Computing. *Baeldung*. 2023. URL: <https://www.baeldung.com/cs/centralized-vs-distributed-computing>
27. Gong S., Ye H., Su P. A Peer-to-Peer Based Distributed Collaborative Filtering Architecture. *International Joint Conference on Artificial Intelligence*, Hainan, China. 2009. P. 305–307. DOI: 10.1109/JCAI.2009.154
28. Karaarslan E., Konacaklı E. Data storage in the decentralized world: Blockchain and derivatives. *arXiv preprint arXiv:2012.10253*. 2020. DOI: 10.48550/arXiv.2012.10253
29. Vergne J. P. Decentralized vs. distributed organization: Blockchain, machine learning and the future of the digital platform. *Organization Theory*. 2020. Vol. 1(4). DOI: 10.1177/2631787720977052
30. Kosalka T. Centralized and decentralized distribution storage systems: decision making through the simulation of centralized and decentralized distribution storage systems. *VDM Verlag Dr. Müller*. 2008. 168 p.
31. Si M., Li Q. Shilling attacks against collaborative recommender systems: a review. *Artificial Intelligence Review*. 2020. Vol. 53, P. 291–319. DOI: 10.1007/s10462-018-9655-x
32. Zhang S., Yin H., Chen T., Huang Z., Nguyen Q. V. H., Cui L. Pipattack: Poisoning federated recommender systems for manipulating item promotion. *In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 2022. P. 1415–1423. DOI: 10.1145/3512728

33. Meleshko Y., Yakymenko M., Semenov S. A Method of Detecting Bot Networks Based on Graph Clustering in the Recommendation System of Social Network. *Computational Linguistics and Intelligent Systems*, CEUR-WS. 2021. Vol. 2870. 2021. P. 1249–1261. URL: <https://ceur-ws.org/Vol-2870/paper92.pdf>
34. Isinkaye F. O., Folajimi Y. O., Ojokoh B. A. Recommendation systems: Principles, methods and evaluation. *Egyptian informatics journal*. 2015. Vol. 16(3). P. 261–273. DOI: 10.3390/electronics11010141
35. Naeen H. M., Jalali M. A decentralized trust-aware collaborative filtering recommender system based on weighted items for social tagging systems”. *Social and Information Networks*. 2019. DOI: 10.48550/arXiv.1906.05143
36. Wang Y. et al. Random Feature-Based Collaborative Kernel Fuzzy Clustering for Distributed Peer-to-Peer Networks. *IEEE Transactions on Fuzzy Systems*. 2023. Vol. 31. No. 2. P. 692–706, DOI: 10.1109/TFUZZ.2022.3188363
37. Wang T., Wang M. Distributed collaborative filtering recommendation algorithm based on DHT. *Cluster Comput.* 2019. Vol. 22 (Suppl 2). P. 2931–2941. DOI: 10.1007/s10586-018-1699-9
38. Zhao Z., Wang Y., Wang H. SDN-based cross-domain cooperative method for trusted nodes recommendation in Mobile crowd sensing. *Peer-to-Peer Netw. Appl.* 2021. Vol. 14. P. 3793–3805. DOI: 10.1007/s12083-021-01217-z.
39. Lisi A., De Salve A., Mori P., Ricci L. A Smart Contract Based Recommender System. *Economics of Grids, Clouds, Systems, and Services. GECON 2019. Lecture Notes in Computer Science* / Djemame K., Altmann J., Bañares J., Agmon Ben-Yehuda O., Naldi M. (eds). 2019. Vol. 11819. Springer, Cham. DOI: 10.1007/978-3-030-36027-6_3
40. Bobadilla D. Lipizzi C. A Blockchain-Based Collaborative Filtering Recommendation System Based on Trust. *18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Chengdu, China. 2021. P. 121–126. DOI: 10.1109/ICCWAMTIP53232.2021.9674128
41. Badis L., Amad M., Aïssani D., Abbar S. P2PCF: A collaborative filtering based recommender system for peer to peer social networks. *Journal of*

High Speed Networks. P. 13–31. 2021. Vol. 27(1). DOI: 10.3233/jhs-210649

42. Yoo H., Chung K. Mining-based lifecare recommendation using peer-to-peer dataset and adaptive decision feedback. *Peer-to-Peer Netw. Appl.* 2018. Vol. 11. P.1309–1320. DOI: 10.1007/s12083-017-0620-2

43. Zeng X., Yu X. P2P based on network behavior analysis trust value calculation method. *Proc. SPIE 12714, International Conference on Computer Network Security and Software Engineering (CNSSE 2023)*, 127140E (26 June 2023). 2023. DOI: 10.1117/12.2683177

44. Sadalage P., Fowler M. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. *Addison-Wesley Professional*. 2012. 192 p.

45. Meier A., Kaufmann M. SQL & NoSQL Databases. Springer Vieweg, Wiesbaden. 2019. P. 201–218. URL: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.468.7089>

46. Cure O., Blin G. RDF Database Systems: Triples Storage and SPARQL Query Processing. *Elsevier Science*. 2014. 256 p.

47. Khan W., Kumar T., Zhang C., Raj K., Roy A. M., Luo B. SQL and NoSQL Database Software Architecture Performance Analysis and Assessments – A Systematic Literature Review. *Big Data and Cognitive Computing*. 2023. Vol. 7(2):97. DOI: 10.3390/bdcc7020097. URL: <https://www.mdpi.com/2504-2289/7/2/97>

48. Chang ML. E., Chua H. N. SQL and NoSQL Database Comparison. In: *Arai K., Kapoor S., Bhatia R. (eds). Advances in Information and Communication Networks. FICC 2018. Advances in Intelligent Systems and Computing*. 2019. Vol 886. Springer, Cham. DOI: 10.1007/978-3-030-03402-3_20

49. Malik A., Burney A., Ahmed F. A comparative study of unstructured data with SQL and NO-SQL database management systems. *Journal of Computer and Communications*. 2020. Vol. 8(4). P. 59–71. DOI: 10.4236/jcc.2020.84005 URL: <https://www.scirp.org/journal/paperinformation.aspx?paperid=99539>

50. Arshad M., Brohi M. N., Soomro T. R., Ghazal T. M., Alzoubi H. M., Alshurideh M. NoSQL: Future of BigData Analytics Characteristics and

Comparison with RDBMS. In: *Alshurideh M., Al Kurdi B. H., Masa'deh R., Alzoubi H.M., Salloum S. (eds). The Effect of Information Technology on Business and Marketing Intelligence Systems. Studies in Computational Intelligence*, Springer, Cham. 2023. Vol. 1056. DOI: 10.1007/978-3-031-12382-5_106

51. Makris A., Tserpes K., Anagnostopoulos D., Nikolaidou M., Fernandes de Macedo J. A. Database system comparison based on spatiotemporal functionality. In: *Proceedings of the 23rd International Database Applications & Engineering Symposium (IDEAS '19)*. Association for Computing Machinery, New York, NY, USA. 2019. Article 21. P. 1–7. DOI: 10.1145/3331076.3331101 URL: <https://dl.acm.org/doi/10.1145/3331076.3331101>

52. Neo4j Documentation. *Official website of the graph database Neo4j*. 2021. URL: <https://neo4j.com/docs/>

53. Meleshko Ye. Computer model of virtual social network with recommendation system. *Scientific journal Innovative Technologies and Scientific Solutions for Industries*, Kharkiv, NURE. 2019. Iss. 2(8). P. 80–84.

54. Kumar A. Implementing real time recommendation systems using graph algorithms & exploring graph analytics in a graph database platform (neo4j). *Dissertation submitted in part fulfilment of the requirements for the degree of Master of Science in Data Analytics at Dublin Business School*. 2019. URL: <https://esource.dbs.ie/handle/10788/3708>

55. Kausar M. A., Nasar M. SQL versus NoSQL databases to assess their appropriateness for big data application. *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*. 2021. Vol. 14(4). P. 1098–1108.

56. Robinson I., Webber J., Eifrem E. *Graph Databases.*, O'Reilly Media. 2013. 178 p.

57. Yi N., Li C., Feng X., Shi M. Design and implementation of movie recommender system based on graph database. *14th Web Information Systems and Applications Conference (WISA)*, IEEE. 2017. P. 132–135.

58. Aho A.V., Hopcroft J.E., Ullman J.D. Data Structures and Algorithms. *Pearson*. 2001. 620 p.
59. Jain H. Data Structures & Algorithms In Go. *Hemant Jain*. 2022. 584 p.
60. Rocca La M. Advanced Algorithms and Data Structures. *Mannin*. 2021. 768 p.
61. Крєневич А.П. Алгоритми і структури даних: підручник. Київ, ВПЦ "Київський Університет". 2021. 200 с.
62. Minato S. Zero-suppressed BDDs and their applications. *International Journal on Software Tools for Technology Transfer*. 2001. №3.2. P. 156–170.
63. Knuth D. Art of Computer Programming, The: Combinatorial Algorithms, Vol. 4A, Part 1. *Addison-Wesley Professional*; 1st edit. 2011. 912 p.

РОЗДІЛ 2

МАТЕМАТИЧНА МОДЕЛЬ ЗБОРУ І ОБРОБКИ ДАНИХ ДЛЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ДЕЦЕНТРАЛІЗОВАНОЇ ОДНОРАНГОВОЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ

У цьому розділі запропоновано математичну модель збору і обробки даних для рекомендаційної системи децентралізованої однорангової комп'ютерної мережі. Математична модель розроблялася для вирішення актуальної задачі підвищення точності математичного моделювання процесів збору та обробки даних у рекомендаційних системах для однорангових децентралізованих комп'ютерних мереж.

Розроблена математична модель збору і обробки даних для рекомендаційної системи, що заснована на GERT-схемі та відрізняється від відомих використанням мультитригерного підходу та врахуванням вимог до достовірності і безпеки даних під час змін у структурі децентралізованої P2P комп'ютерної мережі [1].

Запропонована математична модель може бути використана для прототипування рекомендаційних систем у різних сферах діяльності.

2.1. Матеріали та методи дослідження для розробки математичної моделі збору і обробки даних для рекомендаційної системи децентралізованої однорангової комп'ютерної мережі

Для математичного моделювання процесів збору і обробки даних для рекомендаційної системи однорангової децентралізованої комп'ютерної мережі було обрано підхід GERT-мережевого моделювання [2-4]. GERT моделювання є альтернативним ймовірнісним методом мережевого планування, що використовується, коли існує багатоваріантність реалізації проєкту. І, як правило, воно застосовується для планування інноваційних проєктів.

GERT-мережа представляє собою мережу стохастичної структури, що дозволяє досліджувати ймовірнісно-часові характеристики різних мереж і, зокрема, мереж передачі даних [2-4].

Дослідження у цьому розділі проводилися з використанням системи комп'ютерної алгебри із класу систем автоматизованого проєктування MathCad [5] міжнародної компанії-розробника програмного забезпечення PTC. Для оцінки точності розробленої моделі було використано наближені оцінки, засновані на обробці даних із відкритого набору даних Netflix Prize data [6]. Точність результатів моделювання оцінюється за допомогою показника довірчої ймовірності потрапляння в «усереднений» довірчий інтервал [7-9].

Виконання технології GERT-мережевого моделювання та питання забезпечення точності результатів моделювання потребують ряду попередніх та безпосередніх етапів.

Перший етап – це формування структури, що відповідає вказаним вимогам безпеки і достовірності та визначення вхідних даних. На цьому етапі визначаються основні правила функціонування мережі, її характеристики, такі як часові параметри, ймовірнісні розподіли та інші параметри, що впливають на роботу системи. Другий етап – це побудова GERT-моделі та розрахунок ймовірнісно-часових характеристик. GERT-модель повинна включати події та зв'язки між ними. При цьому потрібно застосовувати математичні методи для розрахунку характеристик мережі, таких як середня тривалість (час) формування рекомендацій, передачі файлу для різних «датасетів» тощо.

Серед основних припущень, що враховуються в роботі, є те, що рекомендаційні системи мають вразливості, які дають змогу злоумисникам отримувати несанкціонований доступ до даних або впливати на рекомендації. Крім того, об'єктивні та суб'єктивні зовнішні фактори є джерелами невизначеності вхідних даних та впливають на достовірність результатів формування рекомендацій.

Основним обмеженням, що розглядається, є формат мережі P2P. Це розповсюджується на такі показники, як масштабованість, автономність вузлів, конективність, надійність та інші.

У цій роботі розглядається чисельний метод знаходження безперервної щільності розподілу ймовірностей часу проходження GERT-мережі за умови, що множина розподілів, якими можуть характеризуватися окремі дуги моделі, включає в себе експоненціальний розподіл.

Запропонований метод ґрунтуються на переході від еквівалентної W -функції $W_E(s)$ GERT-мережі до її характеристичної функції $X_E(\xi)$ і використанні формули обертання [10].

Щільність розподілу ймовірностей часу проходження GERT-мережі визначається таким виразом:

$$\zeta(x) = \frac{1}{2\pi} \int_{-i\infty}^{i\infty} e^{-i\xi x} X_E(\xi) d\xi.$$

Характеристична функція $X_E(\xi)$ знаходиться на основі топологічного рівняння Мейсона [10] шляхом заміни в еквівалентній похідній функції моментів $M_E(s)$ змінної s на $i\xi$, де ξ – дійсна змінна.

Для забезпечення умов інтегрування в підінтегральний вираз введено множник $\exp(-0.5\xi^2)$ [10]. Це рівносильно додаванню в GERT-мережу послідовної гілки, описуваної нормально розподіленою випадковою величиною ζ_2 з нульовим математичним очікуванням і дисперсією, що дорівнює одиниці. Фіктивну гілку можна ввімкнути відразу після джерела s мережі. Якщо випадкова величина ζ_1 є часом проходження GERT-мережі, то щільність розподілу $\zeta(x)$ суми $\zeta_1 + \zeta_2$ визначається виразом $X_E(\xi) = X_E(\xi) \exp(-0.5\xi^2)$.

Після знаходження щільності $\zeta(x)$ має бути знайдена шукана щільність розподілу $\zeta(x)$. Це досягається використанням чисельного методу трансформації законів розподілу на основі розв'язання системи лінійних рівнянь.

2.2. Розробка математичної моделі збору і обробки даних для рекомендаційної системи децентралізованої однорангової комп'ютерної мережі

При моделюванні рекомендаційної системи децентралізованої однорангової комп'ютерної мережі виникає велика кількість завдань, які можуть бути вирішені з використанням різних GERT-моделей. Вони можуть використовуватися як незалежно одна від одної, так і в комбінованих системах. У розпорядження користувача можуть бути надані кілька нових різновидів GERT-моделей, зокрема однорідні мережі великої розмірності, неоднорідні мережі, мережі зі старінням заявок, випадкові GERT-мережі тощо [12, 13].

2.2.1. Структура рекомендаційної системи однорангової децентралізованої комп'ютерної мережі з використанням мультитригерного підходу та врахуванням підвищених вимог до достовірності і безпеки даних

Для визначення входних параметрів GERT-моделі та формування відповідної GERT-мережі, з урахуванням вимог достовірності та безпеки даних у рекомендаційних системах, введемо кілька припущень та пропозицій.

Багатовузловий моніторинг безпеки об'єктів рекомендаційної системи – це процес, метою якого є достовірне визначення стану безпеки об'єкта моніторингу Θ . Такий моніторинг відбувається для виявлення змін в межах існуючого положення в полі станів F вузлів моніторингу та має на меті фіксацію змін стану безпеки p_{Θ} .

Під зміною стану безпеки об'єкта рекомендаційної системи p_{Θ} розумітимемо впорядковану множину пар:

$$p_{\Theta} = \{x_i, t_i\} : \forall i = 1 \dots n, \quad x_i \in F,$$

де x_i – показник, що фіксує стан безпеки об’єкта; t – момент часу спостереження об’єкта моніторингу Θ ; i – номер спостереження об’єкта моніторингу Θ .

Нехай із метою забезпечення децентралізованого виконання функції багатовузлового моніторингу об’єктів рекомендаційної системи внутрішня підсистема аналітичного моніторингу безпеки об’єктів має архітектуру однорангової мережі інтелектуальних вузлів моніторингу, що формують множину R . Для мережевої адресації в такій підсистемі можемо використовувати технологію безкласової IP-адресації. Тоді відображення $C^{(R \rightarrow R)}$ множини R визначає структуру логічних зв’язків елементів цієї множини. При цьому результат процедури призначення IP-адрес з деякої множини A елементам цієї множини R може описуватись бієктивним відображенням: $C^{(R \rightarrow A)}$. Крім того, сюр’єктивне відображення $C^{(R \rightarrow \Psi)}$ в моделі може формалізувати зв’язок елементів множини R з елементами множини їх безпосереднього зв’язку Ψ на площині моніторингу.

Тоді для виконання функції багатовузлового моніторингу безпеки об’єктів рекомендаційної системи елементи множини R повинні реалізовувати механізми доступу та розподіленого зберігання даних, що являють собою множину $\aleph(\psi_r, \partial_r)$, $\psi_r \in \Psi$, $\partial_r \in A$, $r \in R$.

Нехай:

$$\aleph : \{\aleph(\psi_r, \partial_r)\} \xrightarrow{ON} R. \quad (2.1)$$

Тоді $\forall r_i \in R$ використовуючи логічні зв’язки, що задані відображенням $C^{(R \rightarrow R)}$, є можливість додавати $\{\aleph(\psi_r, \partial_r)\}$ у відношення (2.1) і здійснювати пошук:

$$\partial_{r_j} : (x, y) \in \psi_{r_j}, \left((\partial_{r_j}, \psi_{r_j}), r_k \in \aleph \right), \quad (2.2)$$

де (x, y) – аргумент пошукового запиту.

Порядок виконання функції багатовузлового моніторингу безпеки

об'єктів визначається алгоритмом, що формалізується за допомогою функціоналу:

$$L^{(MMO)} = \left\{ \bigcup_{i=1..|R|} L_{r_i}^{(MMO)} : L_{r_i}^{(MMO)} = \dots = L_{r_{|R|}}^{(MMO)} \right\} : \{(MOD, IDS, NET)\} \rightarrow TR, \quad (2.3)$$

де $L_{r_i}^{(MMO)}$ – формалізація алгоритму багатовузлового моніторингу безпеки об'єктів інтелектуальних вузлів r_i рекомендаційної системи; MOD – множина потоків даних з вузлів моніторингу з множини R ; IDS – множина ідентифікаційних ознак безпеки об'єкта моніторингу; NET – однорангова мережа інтелектуальних вузлів моніторингу; TR – множина всіх можливих станів безпеки об'єкта моніторингу.

З метою підвищення достовірності та безпеки даних в рекомендаційних системах встановимо наступний порядок розподіленого зберігання даних в окремих елементах мережі R :

$$\bigcup_{i=1..|R|} S_{r_i} \equiv S,$$

де $\bigcup_{i=1..|R|} S_{r_i}$ – множина районів відповідальності елементів R , що задається

бієктивним відображенням.

Правило закріплення $\{\aleph(\psi_r, \partial_r)\}$ за елементами множини S таке: якщо аргументи $\forall (x_{\psi_{r_i}}, y_{\psi_{r_i}}) \in S_{r_j}$, то механізми доступу та розподіленого зберігання даних $\{\aleph(\psi_{r_i}, \partial_{r_i})\} \xrightarrow{ON} \{r_j\}$.

З метою підвищення достовірності рекомендацій всі елементи мережі R повинні мати можливість здійснювати «інформаційну взаємодію» між собою. Також із метою зниження навантаження на канали зв'язку, що виникає під час роботи рекомендаційної системи, та підвищення її безпеки, відображення $C^{(R \rightarrow R)}$ має бути ін'єктивним.

Проведені дослідження показали, що ця умова може бути досягнута транзитивно – за допомогою композиції відображень логічних зв'язків, тобто

$\forall \{r_i, r_j\}$ повинна виконуватися умова:

$$\exists C^{(r_i \rightarrow r_j)} = \left(C^{(r_i \rightarrow R)} \circ \dots \circ C^{(R \rightarrow r_j)} \right) : \{r_i\} \rightarrow \{r_j\}, \quad (2.4)$$

$$\text{де } \left(C^{(r_i \rightarrow R)} \dots C^{(R \rightarrow r_j)} \right) \subset C^{(R \rightarrow R)}.$$

Припустимо, що структура $C^{(R \rightarrow R)}$ визначається наявністю у елементів S загальних зв'язків. Відношення між такими вузлами рекомендаційної системи, що вважаються «сусідніми», можна описати таким чином:

$$C^{(R \rightarrow R)} = \bigcup_{i=1..|R|} C^{(r_i \rightarrow R)} = \left\{ (r_i, r_j) : s_{r_i} \cap s_{r_j} \right\} : R \rightarrow R, \quad (2.5)$$

де $C^{(r_i \rightarrow R)}$ – частина даних про найближчі зв'язки рекомендаційної системи. Вузли рекомендаційної системи, що знаходяться у такому відношенні, знають ІР-адреси та зони відповідальності один одного.

Тоді відношення, що визначає зв'язок «батьківських» вузлів рекомендаційної системи з «дочірніми» вузлами, інформація про стан яких зберігається у «батьківських» вузлах, може бути записане в наступному вигляді:

$$C_{data}^{(R \rightarrow R)} = \bigcup_{i=1..|R|} C_{data}^{(r_i \rightarrow R)} = \left\{ (r_i, r_j) : s_{r_i} \cap \psi_{r_j} \right\} : R \rightarrow R, \quad (2.6)$$

де $C_{data}^{(r_i \rightarrow R)}$ – локальна інформація «батьківського» вузла r_i про свої «дочірні» вузли. Учасники таких відносин мають дані про адресну інформацію один одного.

На рис. 2.1 представлений приклад структури децентралізованої однорангової мережі з рекомендаційною системою, сформованою з урахуванням підвищених вимог щодо достовірності та безпеки даних шляхом зв'язування зон відповідальності «батьківських» та «дочірніх» вузлів.

Для схематичного представлення майбутнього прототипу моделі рекомендаційної системи децентралізованої однорангової комп'ютерної мережі пропонується використовувати концепцію мобільних агентів.

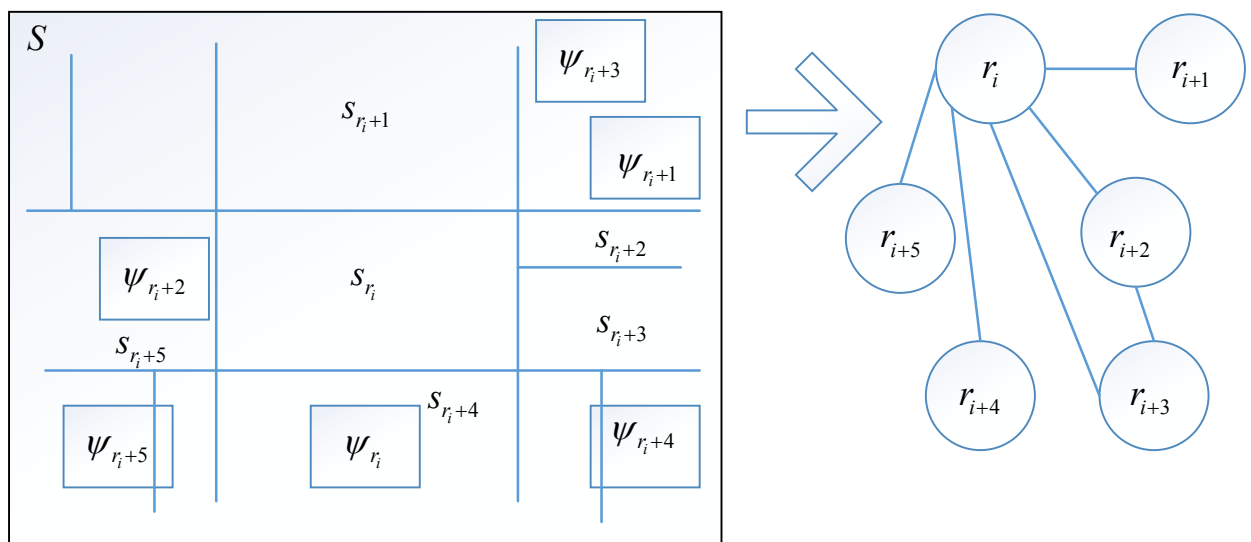


Рис. 2.1. Приклад структури рекомендаційної системи однорангової мережі, сформованої з урахуванням підвищених вимог до достовірності і безпеки даних шляхом зв'язування зон відповідальності «батьківських» та «дочірніх» вузлів

Множина вузлів рекомендаційної системи розглядається як система на основі агентів та тригерів. В системі кожен вузол представлений об'єктом-агентом, що має функції, пов'язані з досягненням локальної мети – відслідковуванням зміни станів об'єктів з наявної множини «дочірніх» об'єктів у межах своєї відповідальності. Тригери є явищами, що сигналізують про зміну структури мережі та вказують про необхідність обробки та врахування цих змін для формування якісних рекомендацій. Крім того, він відповідальний за передачу даних про «дочірній» вузол при наявності таких змін.

З метою підвищення достовірності та безпеки даних, а також децентралізації мультитригерної системи при виконанні функції багатовузлового моніторингу об'єктів інфраструктуру мережі рекомендаційної системи представимо у вигляді, як на рис. 2.2.

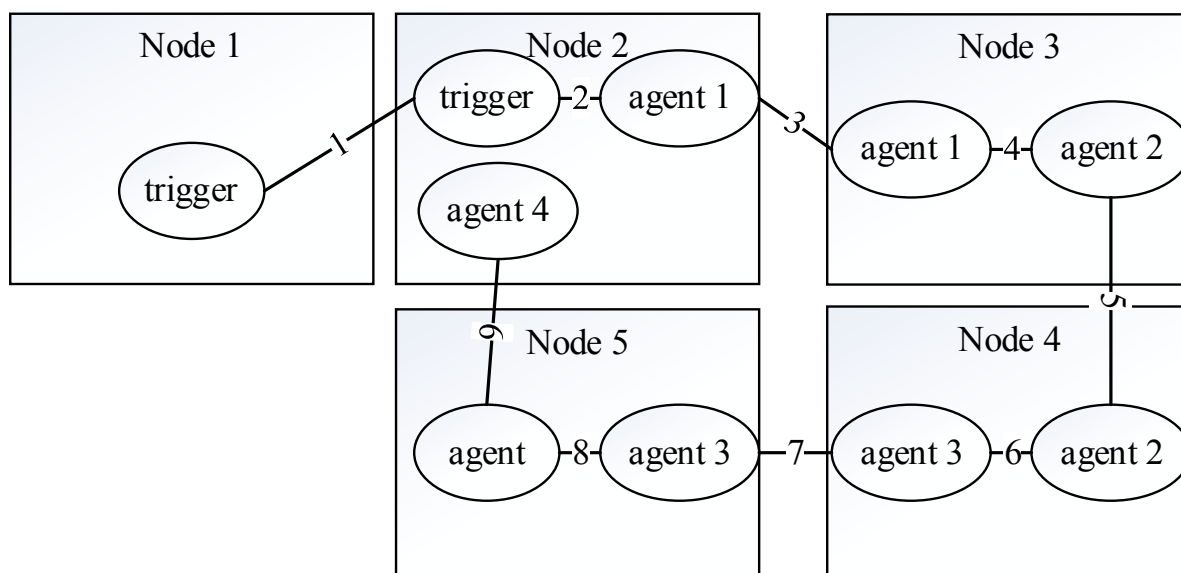


Рис. 2.2. Приклад процесу взаємодії вузлів мультитригерної системи збору та обробки даних для рекомендаційної системи

Така рекомендаційна система складається з частин двох типів: «агент» – програмно-апаратна платформа, що реалізує функції обміну ідентифікаційних даних про стан вузлів рекомендаційної системи, та «тригер» – платформа, що ініціює формування підмножин «дочірніх» та «батьківських» вузлів.

2.2.2. Формалізація та оцінка ймовірно-часових характеристик процесів формування та зміни рекомендацій в одноранговій децентралізованій комп'ютерній мережі

GERT-схему ідентифікації стану вузлів рекомендаційної системи в процесі формування та зміни рекомендацій представлено на рис. 2.3.

У наведеній схемі стан 1 відповідає статусу «Початковий», в якому змін нема. Стан 2 відповідає статусу «Відбулись зміни». В цьому стані в якомусь із вузлів може змінитись рекомендаційний «портрет», тобто може сформуватись нова ідеологія рекомендацій. Стан 4 відповідає статусу «Про зміни повідомлено». В цьому стані вузол «батько» помічає про зміни ідеології

рекомендацій (рекомендаційного «портрета» «дочірнього» вузла). Стан 5 – статус «Перевірено та зафіксовано про зміни стану». В цьому стані фіксується «портрет» «дочірнього» вузла в «батьківському» вузлі. Стан 3 відповідає статусу «Зафіксовано у всіх вузлах». В цьому стані фіксується «портрет» «дочірнього» вузла в інших вузлах.

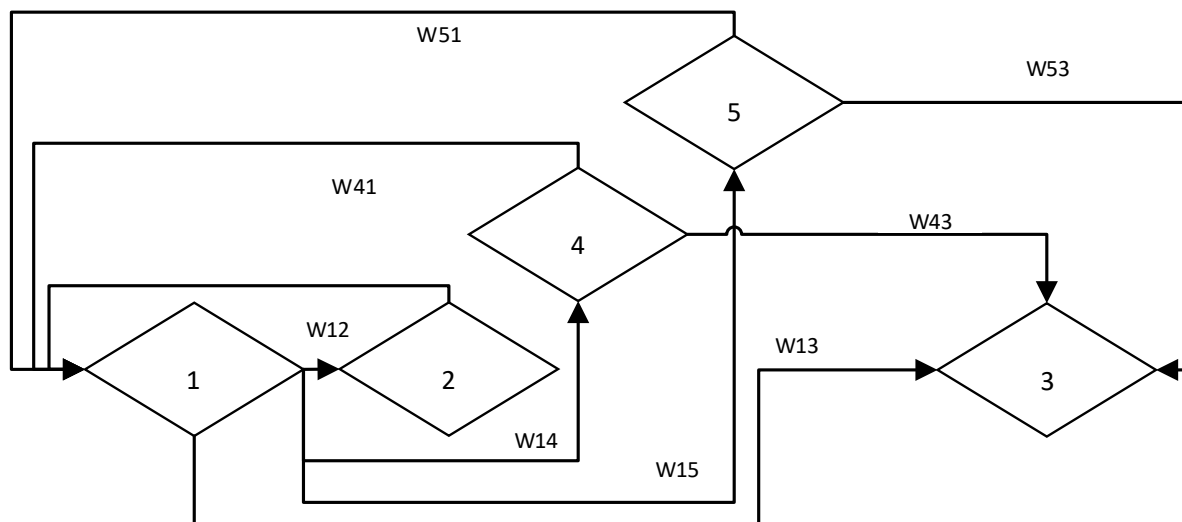


Рис. 2.3. GERT-схема ідентифікації стану вузлів рекомендаційної системи в процесі формування та зміни рекомендацій

Особливостями цієї схеми є врахування станів 4 та 5 в розробленій моделі. Це врахування дозволить, по-перше, відобразити запропоновану структуру рекомендаційної системи однорангової мережі (рис. 2.1) в GERT-моделі. По друге, це дало змогу математично формалізувати комунікативні процеси між «батьківським» та «дочірнім» вузлами. В умовах, коли саме на «батьківський» вузол може бути покладено відповідальність на безпеку та достовірність вхідних даних, таке «поглиблення» GERT-схеми ідентифікації стану вузлів рекомендаційної системи дозволить підвищити точність результатів моделювання.

Характеристики гілок GERT-схеми представлено в табл. 2.1.

Таблиця 2.1. Характеристики гілок GERT-схеми ідентифікації стану вузлів рекомендаційної системи в процесі формування та зміни рекомендацій

<i>W</i>-функція	Ймовірність	Твірна функція моментів
W_{12}	p_1	$\lambda_1/(\lambda_1-s)$
W_{14}	p_2	$\lambda_2/(\lambda_2-s)$
W_{15}	q_1	$\lambda_2/(\lambda_2-s)$
W_{13}	p_3	$\lambda_3/(\lambda_3-s)$
W_{21}	q_2	$\lambda_4/(\lambda_4-s)$
W_{43}	p_4	$\lambda_5/(\lambda_5-s)$
W_{41}	q_3	$\lambda_4/(\lambda_4-s)$
W_{53}	p_5	$\lambda_5/(\lambda_5-s)$
W_{51}	q_4	$\lambda_4/(\lambda_4-s)$

В табл. 2.1 $q_1=1-p_1-p_2$; $q_2=1-p_3$; $q_3=1-p_4$; $q_4=1-p_5$.

Еквівалентна *W*-функція [10, 11] часу ідентифікації стану вузлів рекомендаційної системи в процесі формування та зміни рекомендацій дорівнює:

$$W_E(s) = \frac{W_{13} + W_{14}W_{43} + W_{15}W_{53}}{1 - W_{15}W_{51} - W_{14}W_{41} - W_{12}W_{21}} = \frac{\left(\frac{p_3\lambda_3}{(\lambda_3-s)} + \left(\frac{p_2\lambda_2}{(\lambda_2-s)} \times \frac{p_4\lambda_5}{(\lambda_5-s)} \right) + \left(\frac{p_5\lambda_5}{(\lambda_5-s)} \times \frac{q_1\lambda_2}{(\lambda_2-s)} \right) \right)}{1 - \left(\frac{q_1\lambda_2}{(\lambda_2-s)} \times \frac{q_4\lambda_4}{(\lambda_4-s)} \right) - \left(\frac{p_2\lambda_2}{(\lambda_2-s)} \times \frac{q_3\lambda_4}{(\lambda_4-s)} \right) - \left(\frac{p_1\lambda_1}{(\lambda_1-s)} \times \frac{q_2\lambda_4}{(\lambda_4-s)} \right)} \quad (2.7)$$

Після аналізу виразу (2.7) отримаємо наступне співвідношення:

$$W_E(s) = \frac{(\lambda_4 - s) \times (\lambda_2 \lambda_5 q_5 (\lambda_3 - s) + p_3 \lambda_3 (\lambda_2 - s) (\lambda_5 - s))}{\left((\lambda_3 - s) (\lambda_5 - s) \times \right. \\ \left. \times ((\lambda_2 - s) ((\lambda_4 - s) (\lambda_1 - s) - p_1 q_2 \lambda_1 \lambda_4) - \lambda_2 \lambda_4 q_6 (\lambda_1 - s)) \right)}, \quad (2.8)$$

де $q_5 = (p_2 p_4 + q_1 p_5)$, $q_6 = (q_1 q_4 - q_3 p_2)$.

Недавні дослідження показали, що у складних GERT-мережах з можливими циклами немає простих методів для знаходження особливих точок функції $\mathfrak{I}_E(z)$. Це пов'язано з тим, що для знаходження таких точок потрібно вирішувати нелінійні рівняння, і чим більш складна структура GERT-мережі, тим складніше вирішувати відповідне рівняння. Тому під час моделювання і виконання комплексних перетворень можна замінити дійсні змінні ($z = -i\zeta$), де ζ є дійсною змінною, та отримати наступне:

$$\mathfrak{I}(z) = \frac{bz^3 + az^2 + uz + k}{(\lambda_3 + z)(\lambda_5 + z) \times (z^3 + wz^2 + dz + r)}, \quad (2.9)$$

де:

$$\begin{aligned} b &= q_5 \lambda_5 \lambda_2 + p_3 \lambda_3, \\ a &= q_5 \lambda_5 \lambda_2 (\lambda_3 + \lambda_4) + p_3 \lambda_3 (\lambda_4 + \lambda_2 + \lambda_5), \\ u &= q_5 \lambda_5 \lambda_2 (\lambda_3 \lambda_4 + \lambda_4 + \lambda_3) + p_3 \lambda_3 (\lambda_2 \lambda_4 + \lambda_5 \lambda_4 + \lambda_2 \lambda_5), \\ k &= \lambda_2 \lambda_3 \lambda_4 \lambda_5 (q_5 + p_3), \\ w &= \lambda_1 + \lambda_4 + \lambda_2, \\ d &= \lambda_2 (\lambda_4 + \lambda_1 + \lambda_4 (q_1 q_4 \lambda_1 - q_2 q_3)) + \lambda_1 \lambda_4 (1 - p_1 q_2), \\ r &= \lambda_1 \lambda_4 \lambda_2 (1 - p_1 q_2 - q_1 q_4 + p_2 q_3). \end{aligned}$$

Як особливість вирішення цього рівняння можна виділити наступне. Якщо функція $\mathfrak{I}_E(z)$ в межах $Re z < 0$ задовільняє вимогам леми Жордана, то інтеграл, узятий уздовж контуру Бромвича, являє собою суму вирахувань функції $\mathfrak{I}_E(z)$ щодо всіх її особливостей:

$$\zeta(x) = \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} e^{zx} I_E(z) dz = \sum_{k=1}^n Res[e^{zx} I_E(z)].$$

Для виконання умов леми Жордана необхідно, щоб функція $\mathfrak{I}_E(z)$ у

лівій півплощині була аналітичною за винятком скінченної кількості полюсів і рівномірно відносно аргументу z прямувала до нуля при $|z| \rightarrow \infty$.

Функція $\mathfrak{Z}_i(z) = \frac{\lambda}{\lambda + z}$, що представляє експоненціальний розподіл, рівномірно прагне до нуля за аргументом z за $|z| \rightarrow \infty$. Крім того, функція має простий полюс у точці $z = -\lambda$.

Як особливість моделі можна також відмітити можливість її адаптування до різних функцій розподілу. Наприклад, якщо використовувати в основі розподіл Ерланга, функція $\mathfrak{Z}_i(z) = \frac{\lambda^\alpha}{(\lambda + z)^\alpha}$ рівномірно збігається до нуля залежно від аргументу z при $|z| \rightarrow \infty$. Крім того, в точці $z = -\lambda$ існує полюс кратності α .

Щільність розподілу ймовірностей часу ідентифікації стану вузлів рекомендаційної системи в процесі формування та зміни рекомендацій:

$$\zeta(x) = \frac{1}{2\pi i} \int_{-i\infty}^{i\infty} e^{zx} \frac{bz^3 + az^2 + uz + k}{(\lambda_3 + z)(\lambda_5 + z) \times (z^3 + wz^2 + dz + r)} dz, \quad (2.10)$$

де інтегрування виконується з контуром Бромвіча.

За результатами досліджень встановлено, що метод інтегрування залежить від того, чи функція $\mathfrak{Z}(z)$ має лише прості полюси, чи полюси вищого порядку. Якщо функція $\mathfrak{Z}(z)$ має лише прості полюси, то вираз $e^{zx} \mathfrak{Z}(z)$ можна записати у вигляді:

$$e^{zx} \mathfrak{Z}(z) = \frac{e^{zx} (bz^3 + az^2 + uz + k)}{(z^5 + g_4 z^4 + g_3 z^3 + g_2 z^2 + g_1 z + g_0)} = \frac{\alpha(z)}{\beta(z)}, \quad (2.11)$$

де:

$$\begin{aligned} g_4 &= \lambda_5 + \lambda_3 + w, \\ g_3 &= \lambda_5 \lambda_3 + w(\lambda_5 + \lambda_3) + d, \\ g_2 &= \lambda_5 \lambda_3 w + d(\lambda_5 + \lambda_3) + r, \\ g_1 &= \lambda_5 \lambda_3 + r(\lambda_5 + \lambda_3), \end{aligned}$$

$$g_0 = \lambda_5 \lambda_3 r.$$

Тоді щільність розподілу ймовірностей часу ідентифікації стану вузлів рекомендаційної системи в процесі формування та зміни рекомендацій:

$$\zeta(x) = \sum_{n=1}^5 \text{Res} \left[e^{zx} \mathfrak{Z}(z) \right] = \sum_{n=1}^5 \frac{\alpha(z_n)}{\beta(z_n)} = \sum_{n=1}^5 \frac{e^{z_n x} (z_n^3 b + z_n^2 a + z_n u + k)}{(5z_n^4 + 4g_4 z_n^3 + 3g_3 z_n^2 + 2g_2 z_n + g_1)}. \quad (2.12)$$

Функція $\mathfrak{Z}(z)$ може мати полюси вищого порядку, окрім простих полюсів, які визначаються коренями рівняння $z^3 + wz^2 + dz + r = 0$. Це можливо, якщо значення λ_3 і λ_5 збігаються або дорівнюють значенням коренів z_3, z_4, z_5 . У таких випадках щільність розподілу часу $\zeta(x)$ ідентифікації стану вузлів рекомендаційної системи в процесі формування та зміни рекомендацій можна знайти за формулою, яка використовує полюси z_n порядку m :

$$\gamma_{-1} = \frac{1}{(m-1)!} \lim_{z \rightarrow z_n} \frac{d^{m-1} \left[(z - z_n)^m e^{zx} \mathfrak{Z}(z) \right]}{dz^{m-1}}.$$

Вираз (2.9) можна записати як функцію, що складається з чисельника та знаменника з більш високим ступенем, ніж у чисельника. Це дозволяє використовувати умови леми Жордана для цієї функції. Функція $\mathfrak{z}(z)$ містить полюси у точках $z_1 = -\lambda_3$, $z_2 = -\lambda_5$, а також ще три полюси, що породжені багаточленом $z^3 + wz^2 + dz + r$. Для знаходження розв'язку рівняння використали:

$$z^3 + wz^2 + dz + r = 0, \quad (2.13)$$

та отримали ще три особливі точки z_3, z_4, z_5 .

Як можна помітити, коефіцієнти рівняння (2.13) більше нуля. Тому, якщо це рівняння має дійсні корені, то всі вони від'ємні.

Якщо рівняння (2.13) має від'ємні дійсні корені, то вони обов'язково не другої або третьої кратності. Якщо припустити, що багаточлен $z^3 + wz^2 + dz + r$ породжує полюс третього порядку в точці $z = -1$, то $(z + 1)^3 = z^3 + 3tz^2 + 3t^2z + t^3$. Так як $t < 0$, то $t^3 < 0$, що суперечить умові $t^3 = c = \lambda_1 \lambda_4 \lambda_2 (1 - p_1 q_2 - q_1 q_4 + p_2 q_3) > 0$. Отже, багаточлен $z^3 + wz^2 + dz + r$ не може породжувати полюсів третього порядку.

Можна припустити, що $z^3 + wz^2 + dz + r$ породжує полюс другого порядку в точці $z = -\iota_1$ і полюс першого порядку в точці $z = -\iota_2$, то має виконуватися співвідношення:

$$(z + \alpha_1)^2 (z + \alpha_2) = z^3 + (2\iota_1 + \iota_2)z^2 + (2\iota_1^2 + 2\iota_1\iota_2)z + \iota_1^2\iota_2. \quad (2.14)$$

Враховуючи, що $\iota_2 < 0$ та $\iota_1^2 > 0$ то $\iota_1^2\iota_2 < 0$.

Якщо провести аналіз, то стає зрозуміло, що існує невідповідність, так як $\iota_1^2\iota_2 = c = \lambda_1\lambda_4\lambda_2(1 - p_1q_2 - q_1q_4 + p_2q_3) > 0$. Отже, вираз (2.9) не може породжувати полюсів другого порядку.

Відповідно до основної теореми алгебри будь-яка ціла раціональна функція n -го ступеня має n нулів. Отже, багаточлен $z^3 + wz^2 + dz + r$ може мати або три різних негативних дійсних нулі, або один від'ємний дійсний нуль і два комплексних сполучених нулі.

Використовуючи зазначені постулати, можна знайти одну з ключових характеристик якості рекомендаційної системи – час ідентифікації стану вузлів рекомендаційної системи в процесі формування та зміни рекомендацій.

Таким чином, розроблена GERT-модель процесів збору і обробки даних для рекомендаційної системи децентралізованої однорангової комп'ютерної мережі. Математична модель відрізняється від відомих урахуванням вимог до достовірності та безпеки інформації рекомендаційної системи під час змін у структурі децентралізованої однорангової мережі.

Запропонована модель може бути використана для дослідження процесів формування та зміни рекомендацій користувачам децентралізованих однорангових комп'ютерних мереж. Вона може бути використана також при розробці нових методів, алгоритмів та способів організації рекомендаційних систем, що використовують децентралізовану однорангову структуру комп'ютерної мережі.

2.2.3. Порівняльні дослідження запропонованої GERT-моделі рекомендаційної системи для децентралізованої однорангової комп'ютерної мережі

Для знаходження щільності розподілу ймовірностей часу ідентифікації стану вузлів рекомендаційної системи в процесі формування та зміни рекомендацій $\xi(x)$ визначальними характеристиками моделі є: $\lambda_1=0,4$, $\lambda_2=0,2$, $\lambda_3=0,8$, $\lambda_4=0,3$, $\lambda_5=0,99$, $p_1=0,92$, $p_2=0,01$, $p_3=0,9$, $p_4=0,6$, $p_5=0,5$.

За допомогою виразів (2.11), (2.12) знайшли, що $w=0,9$, $d=0,247$, $r=0,022$.

Розв'язавши рівняння, що стоїть в знаменнику виразу (2.10), за методом Вієта-Кардано, визначили, що функція $\Im(z)$ має прості полюси: $z_1=-0,8$, $z_2=-0,9$, $z_3=-0,483$.

Крім дійсних коренів, є два комплексно-сполучених: $z_4=-0,208+i \times 0,04$, та $z_5=-0,208-i \times 0,04$.

Відповідно до співвідношення (2.12) $\xi(x)$ виглядає наступним чином:

$$\begin{aligned} \sum_{n=1}^5 \text{Res} \left[e^{zx} \Im(z) \right] = \\ = \frac{e^{(\beta+\delta i)_n x} \left((\beta+\delta i)^3 b + (\beta+\delta i)^2 a + (\beta+\delta i) u + k \right)}{\left(5(\beta+\delta i)^4 + 4g_4(\beta+\delta i)^3 + 3g_3(\beta+\delta i)^2 + 2g_2(\beta+\delta i) + g_1 \right)} + \\ + \frac{e^{(\beta-\delta i)_n x} \left((\beta-\delta i)^3 b + (\beta-\delta i)^2 a + (\beta-\delta i) u + k \right)}{\left(5(\beta-\delta i)^4 + 4g_4(\beta-\delta i)^3 + 3g_3(\beta-\delta i)^2 + 2g_2(\beta-\delta i) + g_1 \right)}. \end{aligned} \quad (2.15)$$

Сума значень будь-якої дрібно-раціональної функції:

$$f(z) = \frac{\varpi_m z^m + \varpi_{m-1} z^{m-1} + \dots + \varpi_1 z + \varpi_0}{\ell_m z^m + \ell_{m-1} z^{m-1} + \dots + \ell_1 z + \ell_0}, \quad \varpi_m \neq 0, \quad \ell_m \neq 0,$$

для значень комплексних аргументів, може бути представлена у вигляді

$$\frac{\tau + i\partial}{\wp + i\hbar} + \frac{\tau - i\partial}{\wp - i\hbar}, \quad \text{де } \tau, \partial, \wp, \hbar - \text{ деякі коефіцієнти.}$$

Це положення можна довести методом математичної індукції. Після цього, використовуючи формули Ейлера, можна отримати:

$$\begin{aligned} \sum_{n=1}^5 \operatorname{Res} \left[e^{zx} \Im(z) \right] &= e^{(\beta+\delta i)} \frac{\tau + i\partial}{\wp + i\hbar} + e^{(\beta-\delta i)} \frac{\tau - i\partial}{\wp - i\hbar} = \\ &= \frac{2e^{\beta x}}{\wp^2 + \hbar^2} \left[(\tau\wp + \hbar) \cos(\delta x) + (\tau\wp - \hbar) \sin(\delta x) \right]. \end{aligned} \quad (2.16)$$

де:

$$\begin{aligned} \tau &= \beta b - 3\beta\delta^2 b + \beta^2 a - \delta^2 a + \beta u + k, \\ \partial &= 3\beta^2 \delta b - \delta^3 b + 2\beta\delta a + \delta u, \\ \wp &= 5\beta^4 - 30\beta^2 \delta^2 + 5\delta^4 + 4\beta^3 - 12\beta\delta^2 + 3\beta^2 w - 3\delta^2 w + 2\beta d + r, \\ \hbar &= 20\beta^3 \delta - 20\beta\delta^3 + 12\beta^2 \delta - 4\delta^3 + 6\beta\delta w + 2\delta d. \end{aligned}$$

З урахуванням вищенаведеного функція розподілу ймовірностей часу ідентифікації стану вузлів рекомендаційної системи в процесі формування та зміни рекомендацій визначається як:

$$\zeta(x) = \frac{2e^{-0,208x} \left[\frac{(9,224 \cos(0,043x))}{10^4} + \frac{7,922 \sin(0,043x)}{10^4} \right]}{\frac{1,321}{10^{-3}}}.$$

Цей вираз є зручним для подальшого аналізу процесів ідентифікації стану вузлів рекомендаційної системи під час формування та зміни рекомендацій з врахуванням співвідношення передування операцій.

У табл. 2.2 наведені значення щільності розподілу ймовірностей часу ідентифікації стану вузлів рекомендаційної системи в процесі формування та зміни рекомендацій.

Отримання наведених значень було здійснене за допомогою обчислення $\xi_b(x)$, з використанням багаточлена Лагранжа другого ступеня, який був побудований шляхом інтерполяції підінтегральної функції. Точність обчислення значень $\xi_b(x)$ залежить від точності визначення коренів z_3, z_4, z_5 , при цьому використовувалась та ж сама щільність $\xi_u(x)$.

Метод інтерполяції багаточленом Лагранжа другого ступеня має свою похибку, яка впливає на точність обчислення функції $\xi_u(x)$. Але абсолютна похибка обчислення значень щільності не перевищує 0,002, що достатньо для більшості практичних застосувань, пов'язаних із поставленою метою роботи. Проте для випадків, де вимагається більша точність, можна підвищити точність обчислень, переходячи до інтерполяційних багаточленів вищих ступенів.

Таблиця 2.2. Значення щільності розподілу ймовірностей часу ідентифікації стану вузлів рекомендаційної системи в процесі формування та зміни рекомендацій, отримані методами обчислень та інтерполяції

№	$\xi_v(x)$	$\xi_u(x)$
1	0,281	0,294
2	0,21	0,223
3	0,194	0,207
4	0,178	0,191
5	0,16	0,173
6	0,143	0,156
7	0,126	0,139
8	0,109	0,122
9	0,093	0,106
10	0,078	0,091
11	0,064	0,077
12	0,05	0,063
13	0,038	0,051
14	0,036	0,049
15	0,035	0,048
16	0,033	0,046
17	0,028	0,041
18	0,027	0,04
19	0,026	0,039

Щільність розподілу часу передачі файлу наведена на рис. 2.4.

З графіка на рис. 2.4 видно, що найвищі значення щільності розподілу ймовірностей часу ідентифікації стану вузлів рекомендаційної системи в

процесі формування та зміни рекомендацій припадають на інтервал від 1 до 10 одиниць виміру.

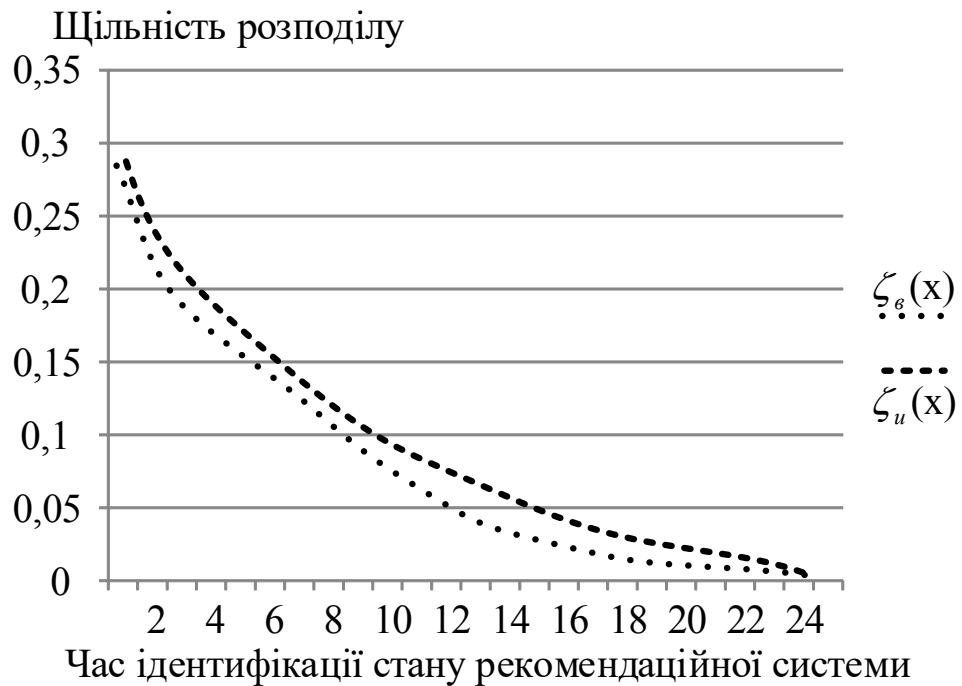


Рис. 2.4. Щільність розподілу ймовірностей часу ідентифікації стану вузлів рекомендаційної системи в процесі формування та зміни рекомендацій

Теоретичні дослідження та практичний досвід адміністрування рекомендаційних систем свідчать про те, що точна кількісна оцінка достовірності та безпеки рекомендаційних повідомлень є неможливою через деяку невизначеність вхідних даних.

Оцінка точності математичної моделі може бути здійснена за допомогою порівняння її прогнозів з реальними даними або експериментальними результатами. Якщо розроблена модель добре передбачає результати, що спостерігаються на практиці, то можна стверджувати, що вона точна. Крім того, можна використовувати різні статистичні методи для порівняння результатів прогнозування з реальними

даними, такі як кореляційний аналіз, метод середньоквадратичного відхилення та інші.

Для оцінки точності розробленої моделі було використано наближені оцінки, засновані на обробці даних із датасету Netflix Prize data [6]. У роботі як основа для оцінки достовірності використовується відомий підхід математичної статистики, заснований на перевірці за критерієм згоди χ^2 Пірсона.

Результати дослідження точності вказують на те, що значення часу ідентифікації стану вузлів рекомендаційної системи в процесі формування та зміни рекомендацій потрапляє з довірчою ймовірністю 0,94 в «усереднений» довірчий інтервал (рис. 2.5). Крім того, результати порівняльного аналізу розробленої моделі з відомими моделями [10, 11] показують покращення цього показника точності до 5%.

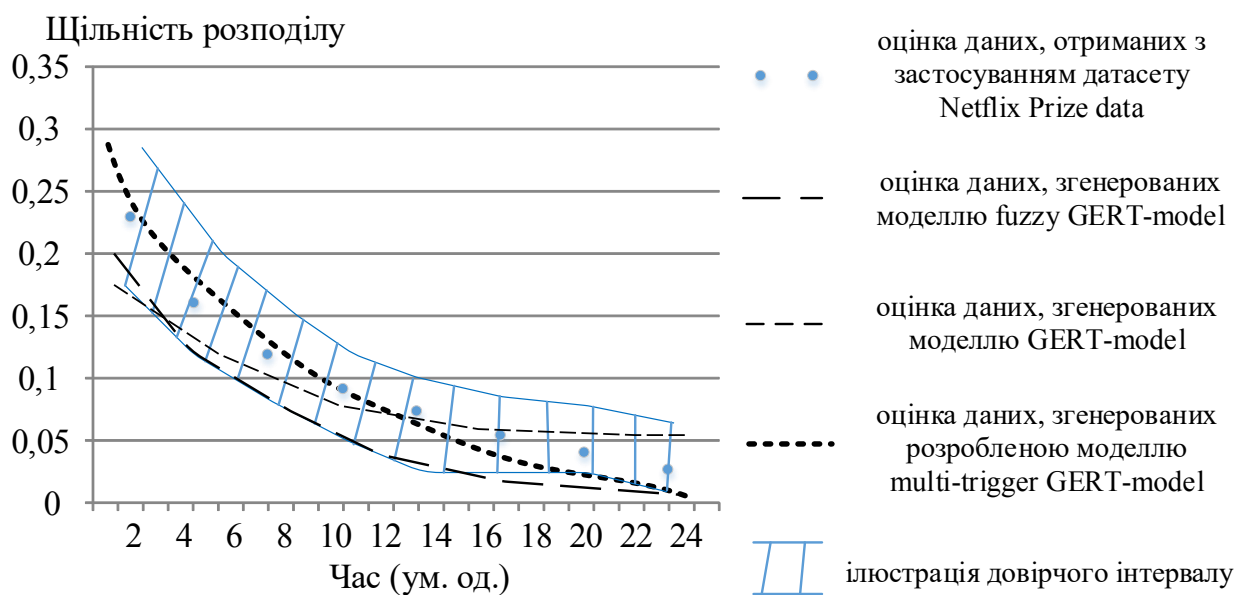


Рис. 2.5. Графіки залежності щільності розподілу від часу передачі файлу для різних «датасетів»

2.2.4. Аналіз результатів та пропозиції щодо подальшого розвитку розробки математичної моделі збору і обробки даних для рекомендаційної системи децентралізованої однорангової комп'ютерної мережі

Розроблена модель децентралізованої рекомендаційної системи дозволила оцінити ймовірно-технічні характеристики процесу обробки даних в рекомендаційній системі децентралізованої однорангової комп'ютерної мережі.

В основі структури рекомендаційної системи, що розроблена, було використано парадигму функціонування мультитригерної однорангової мережі *CAN*. При цьому основною інноваційною складовою цієї розробки стала побудова структури, в якій множина моніторингових вузлів розглядається як мультитригерна система. В цій структурі кожен вузол представлений агентом-об'єктом, що має функції, пов'язані з відслідковуванням та ідентифікацією зміни стану об'єктів моніторингу в межах свого поле зору. З метою досягнення автономності об'єктів-агентів та децентралізації мультитригерної системи для виконання функції багатовузлового моніторингу об'єктів розроблена структура рекомендаційної системи однорангової мережі рис. 2.1.

Одним із основних результатів формалізації процесів формування та змін рекомендацій є отримання аналітичного виразу (2.12). За його допомогою є можливість оцінити щільність розподілу ймовірностей часу ідентифікації стану вузлів рекомендаційної системи. В свою чергу, результати дослідження та оцінки надають можливість визначитись з іншими характеристиками – наприклад, із математичним очікуванням часу ідентифікації стану вузлів рекомендаційної системи або часу передачі файлу для різних «датасетів». Вирішення рівняння (2.12) здійснюється за допомогою виразів (2.13), (2.14). Практичний приклад застосування розробленої GERT-моделі наведено за допомогою рівнянь (2.15), (2.16).

Також однією з переваг цього аналітичного виразу є можливість його використання для різних «датасетів» рекомендаційних систем. Потрібно лише емпіричним шляхом отримати значення основних характеристик (λ та p) GERT-мережі.

Врахування вимог до достовірності та безпеки рекомендаційних повідомлень в порівнянні з відомими моделями дало змогу підвищити точність результатів моделювання. Результати оцінки та дослідження показують покращення показника точності до 5%. Це доведено за допомогою порівняльного аналізу розробленої моделі з відомими моделями [10, 11].

Таке врахування виконано шляхом введення в загальну модель децентралізованої рекомендаційної системи наступних показників – ймовірність достовірної передачі пакета та ймовірність безпечної передачі пакета. Це надає можливість проводити аналіз різних методів рекомендаційних систем з точки зору достовірності та інформаційної безпеки та встановлювати припустимі параметри, при яких ступінь актуальності не падає нижче визначеного порогу.

Отримані результати при порівняльних дослідженнях GERT-моделі можна пояснити коректним обранням показників ефективності роботи рекомендаційної системи, а саме – ймовірності достовірної передачі пакета та ймовірності безпечної передачі пакета. Крім того, імплементація перевіреного способу моделювання за допомогою GERT-мереж у вигляді запропонованої схеми на рис. 2.3 також дала змогу покращити якість отриманих результатів.

Як показали результати досліджень, незважаючи на очевидні переваги запропонованої моделі, існують деякі шляхи, щодо її вдосконалення. Особливо це стосується визначення вхідних характеристик моделі та обранням експоненціального розподілу в практичній частині досліджень. Обрання інших законів розподілу при описі окремих переходів зі стану в стан в подальшому дасть можливість покращити точність результатів моделювання.

Частково в роботі ці недоліки вже усунуті шляхом використання

підходу GERT-мережевого моделювання.

Запропонована GERT-мережева модель також має обмеження. Звісно, що незначне збільшення її елементів-вузлів призведе до значного ускладнення аналітичного виразу (2.7). Тому при моделюванні потрібно використовувати технології раціонального представлення GERT-схем.

Подальший розвиток цього питання можливий за допомогою наступних пропозицій:

1. Розширення функціональності: можна розширити функціональні можливості системи, додавши нові параметри та алгоритми. Наприклад, можна дослідити ефективність використання глибокого навчання або нейронних мереж для покращення точності рекомендацій.

2. Оцінка стійкості моделі: можна дослідити, наскільки стійкою є модель до змін у вхідних даних. Наприклад, можна провести аналіз впливу різноманітних зовнішніх факторів, зокрема, кібервтручання або виходу з ладу окремих компонентів системи.

3. Проведення додаткових експериментів: можна провести експерименти для перевірки точності моделі в різних умовах. Наприклад, можна провести експерименти на різних типах вхідних даних, використавши різні датасети.

Висновки до розділу 2

У цьому розділі запропоновано математичну модель збору і обробки даних для рекомендаційної системи децентралізованої однорангової комп'ютерної мережі. Основною особливістю розробленої моделі є врахування вимог достовірності та безпеки рекомендаційних повідомлень. Це виконано шляхом введення в загальну модель децентралізованої рекомендаційної системи наступних показників: ймовірність достовірної передачі пакета та ймовірність безпечної передачі пакета. Розроблена модель дозволяє проводити порівняльний аналіз різних методів роботи

рекомендаційних систем та встановлювати припустимі параметри, при яких ступінь актуальності не падає нижче визначеного порогу.

Сформовано структуру рекомендаційної системи однорангової децентралізованої комп'ютерної мережі. В основі запропонованої структури представлено мультитригерну систему, в якій кожен вузол представлений об'єктом-агентом. Основною метою такої структури є відслідковування зміни станів об'єктів з наявної множини «дочірніх» об'єктів у межах своєї відповідальності з урахуванням показників достовірності та безпеки.

Формалізовано та оцінено ймовірісно-часові характеристики процесу формування та зміни рекомендацій. Математична формалізація динаміки досліджуваних процесів відрізняється від відомих урахуванням вимог достовірності та безпеки рекомендаційних даних під час змін у структурі однорангової мережі рекомендаційної системи. Це дозволило підвищити точність оцінки ймовірісно-часових характеристик процесів формування та зміни рекомендацій.

Проведено порівняльні дослідження GERT-моделі рекомендаційної системи для децентралізованої однорангової комп'ютерної мережі. Результати дослідження точності вказують на те, що значення часу ідентифікації стану вузлів рекомендаційної системи в процесі формування та зміни рекомендацій потрапляє з довірчою ймовірністю 0,94 в «усереднений» довірчий інтервал. Крім того, результати порівняльного аналізу розробленої моделі з відомими моделями показують покращення цього показника точності до 5%.

Список літератури до розділу 2

1. Mikhav V., Semenov S., Meleshko Y., Yakymenko M., Shulika Y. Constructng the mathematical model of a recommender system for decentralized peer-to-peer computer networks. *Eastern-European Journal of Enterprise Technologies*. 2023. Vol. 4 (9(124)). P. 24–35. (ISSN 1729-3774).

DOI: 10.15587/1729-4061.2023.286187. URL: <https://journals.uran.ua/eejet/article/view/286187> (SCOPUS)

2. Neumann K., Steinhardt U. GERT networks and the time-oriented evaluation of projects. *Springer Science & Business Media*. 2012. Vol. 172.

3. Nelson R. G., Azaron A., Aref S. The use of a GERT based method to model concurrent product development processes. *European Journal of Operational Research*. 2016. Vol. 250. Iss. 2, P. 566–578. DOI: 10.1016/j.ejor.2015.09.040. URL: <https://www.sciencedirect.com/science/article/pii/S0377221715008826>

4. Wyrozębski P., Wyrozębska A. Challenges of project planning in the probabilistic approach using PERT, GERT and Monte Carlo. *Journal of Management and Marketing*. 2013. Vol. 1. Iss. 1. P. 1–8.

5. PTC Mathcad Official website of engineering mathematics software «Mathcad». 2023. URL: <https://www.mathcad.com/>

6. Netflix. Netflix Prize data: Dataset from Netflix's competition to improve their recommendation algorithm. *Kaggle, data mining contest organization system*. URL: <https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data?resource=download>

7. Kiefer J. Conditional Confidence Statements and Confidence Estimators (with discussion). *Journal of the American Statistical Association*. 1977. Vol. 72 (360a): P. 789–827. DOI: 10.1080/01621459.1977.10479956

8. Smithson M. Confidence intervals. *Quantitative Applications in the Social Sciences Series*, Belmont, CA: SAGE Publications. 2003. No. 140.

9. Сеньо П. С. Теорія ймовірностей та математична статистика (вид. 2-ге, перероб. і доп.). Київ: Знання, 2007. 446 с.

10. Semenov S., Zhang L., Cao W., Bulba S., Babenko V., Davydov V. Development of a fuzzy GERT-model for investigating common software vulnerabilities. *Eastern-European Journal of Enterprise Technologies*. 2021. Vol. 6(2(114)). P. 6–18. DOI: 10.15587/1729-4061.2021.243715

11. Semenov S., Liqiang Z., Weiling C., Davydov V. Development a mathematical model for the software security testing first stage. *Eastern-European Journal of Enterprise Technologies*. 2021. Vol. 3(2(111)). P. 24–34. DOI: 10.15587/1729-4061.2021.233417

12. Коваленко О. В. Моделі та методи розроблення безпечного програмного забезпечення комп'ютерних систем : дис. ... д-ра техн. наук : 05.13.05. Черкаси, 2020. 250 с.

13. Богданова Н. В., Богданов О. В. Математичне моделювання систем і процесів. Конспект лекцій : навчальний посібник. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського». 2022. 85 с.

РОЗДІЛ 3

МЕТОД ЗБЕРІГАННЯ ДАНИХ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ, АДАПТОВАНИЙ ДО АРХІТЕКТУРИ ОДНОРАНГОВИХ ДЕЦЕНТРАЛІЗОВАНИХ КОМП'ЮТЕРНИХ МЕРЕЖ

У цьому розділі проведено дослідження методів збереження даних у рекомендаційних системах. Виконано дослідження різних структур даних для зберігання інформації рекомендаційної системи та порівняльний аналіз ефективності їх використання за затратами часу та пам'яті [1-9]. Для проведення експериментів з порівняння ефективності застосування різних структур даних за затратами часу та пам'яті було розроблено програмну імітаційну модель інформаційних процесів [10-16] рекомендаційної системи у комп'ютерній мережі, в якій у цій роботі було окремо виділено три основні сутності: агент, сесія та об'єкт. Проведено серію експериментів на розробленій програмній моделі, зокрема досліджено такі структури даних, як зв'язний список, розгорнутий зв'язний список, хеш-таблиця, В-дерево, В+-дерево та бінарна діаграма рішень.

Вибір методу представлення даних, якими оперує рекомендаційна система, має важливе значення, оскільки ефективний спосіб побудови бази даних для роботи такої системи може зменшити кількість потрібних ресурсів та збільшити кількість доступних алгоритмів для формування списків рекомендацій, а також є важливим з точки зору якості її роботи, швидкості, можливостей масштабування та зручності виконання основних операцій з даними для формування рекомендацій.

Розроблено метод зберігання даних рекомендаційної системи для однорангових комп'ютерних мереж на основі хеш-таблиць та розгорнутих зв'язних списків та проведено експерименти для визначення ефективності його роботи і порівняння з існуючими методами [1-9].

3.1. Розробка методу програмного імітаційного моделювання рекомендаційної системи для комп'ютерних мереж різних архітектур

Імітаційне програмне моделювання рекомендаційних мереж потребує збереження інформації про велику кількість користувачів і об'єктів та зв'язків між ними, тому для їхнього представлення у пам'яті комп'ютера необхідно використовувати структуру даних, яка дуже економно витрачає пам'ять.

Один із найпоширеніших способів представлення графа $G = (V, E)$ – це матриця суміжності. Припустимо, що множина вершин $V = \{1, 2, \dots, n\}$. Матриця суміжності для незваженого графа G – це матриця A розміру $n \times n$ зі значеннями булевого типу, де $A[i, j] = \text{true}$ тоді і тільки тоді, коли існує дуга від вузла i до вузла j [17] (рис. 3.1). На практиці у матрицях суміжності значення true замінюється на 1, а значення false – на 0. Якщо граф зважений, то його також можна представити за допомогою матриці суміжності, але у якій елемент $A[i, j]$ дорівнює вазі дуги $i \rightarrow j$.

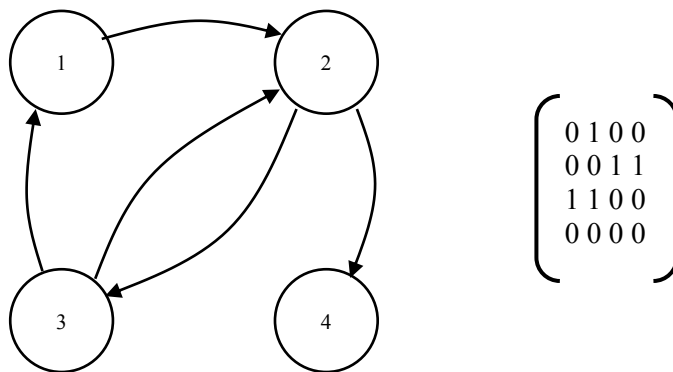


Рис. 3.1. Орієнтований граф та його матриця суміжності

Представлення орграфа у вигляді матриці суміжності зручно застосовувати у тих алгоритмах, у яких необхідно часто перевіряти існування даної дуги. Головний недолік цього підходу в тому, що вона потребує $\Omega(n^2)$

об'єму пам'яті, а для її читання необхідний час порядку $O(n^2)$, незалежно від кількості дуг [17, с. 185].

3.3.1. Представлення орієнтованих графів за допомогою бінарних діаграм рішень у програмній моделі рекомендаційної системи

Матриця суміжності A містить значення булевого типу, тобто вона є матричним представленням булевої функції F , яка описує наявність дуги між двома вузлами.

Як параметри функції F ми пропонуємо використати номери вузлів у бінарній формі. Тоді функція $F(i_0, i_1, \dots, i_m, j_0, j_1, \dots, j_m) = true$, якщо існує дуга $i \rightarrow j$, де $n \in N$, $i, j \in [0, n]$, i_p, j_p – значення біту на позиції p , $p \in [0, m]$, $m = \lfloor \log_2 n \rfloor$.

Якщо орграф G помічений, то необхідно також зберегти його вагу. Для збереження ваги k дуги $i \rightarrow j$ можна використати додаткову групу параметрів $k_0, k_1, \dots, k_l$, $l = \lfloor \log_2(\max k) \rfloor$. У цьому випадку для знаходження ваги дуги необхідний час порядку $O(2^{l+1})$.

Представлення булевої функції як таблиці істинності чи досконалої кон'юнктивної/диз'юнктивної нормальної форми потребує $\Omega(2^{2m+l})$ пам'яті та потребує значних обчислень для визначення значення функції. Тому для представлення булевої функції ми пропонуємо використати бінарні діаграми рішень.

Бінарні діаграми рішень (БДР, binary decision diagram, BDD) – це економна форма представлення булевих функцій у вигляді орієнтованого ациклічного графа [18-20]. Вершини графа представляють аргументи функції, листки – її двійкові значення. При представленні булевих функцій у формі БДР стало можливим розв'язувати багато проблем, які при традиційних представленнях структур нерозв'язні через значну розмірність таких представлень і складність операцій над ними. БДР можуть успішно

застосовуватися фактично в кожній галузі, де потрібно обробляти дискретні структури даних, у тому числі і в комбінаториці. Бінарна діаграма рішень представляє булеву функцію у вигляді мінімального ациклічного направленого графа (3.2).

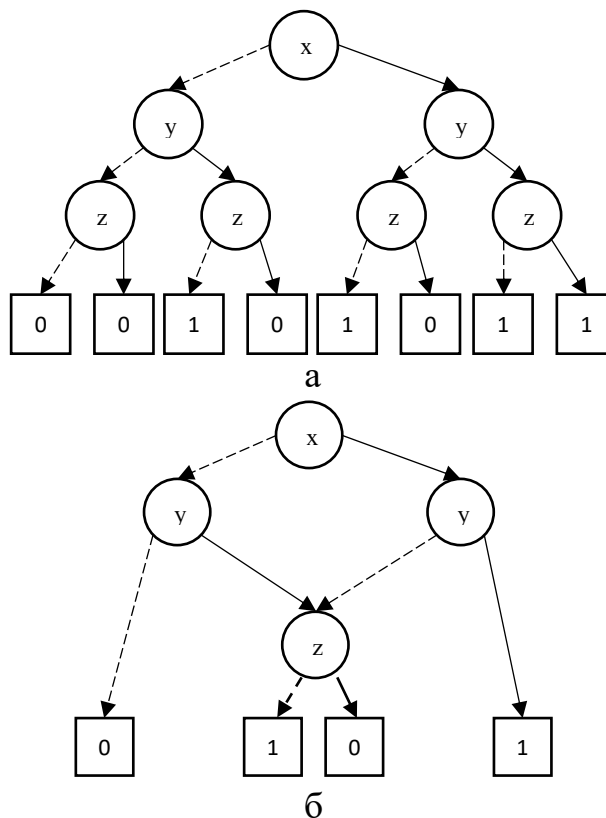


Рис. 3.2 Перехід від (а) бінарного дерева рішень до (б) бінарної діаграми рішень

Адаптація бінарних діаграм рішень для представлення орієнтованих графів здійснювалася наступним чином.

Для додавання і вилучення ребер та зміни ваги ребер необхідно мати можливість редагувати дані графа. БДР дають можливість зберігати дані у стисненому вигляді та швидко отримувати значення функції за її параметрами, але редагування БДР вимагає складних обчислень.

Якщо алгоритм передбачає періоди інтенсивного редагування графа, можна тимчасово зберігати зміни у структурі даних, більш пристосованій до внесення змін. Пропонується застосувати ідею «гарячого» (хеш-таблиця) та

«холодного» (БДР) сховищ. У хеш-таблицю заносяться параметри функції, на яких необхідно змінити значення, та саме значення. При досягненні певного розміру або по завершенню логічного блоку операцій із хеш-таблиці формується коригувальна БДР, яка потім зливається із основною. При читанні значень необхідно спочатку перевірити існування значення у хеш-таблиці, і, у випадку його відсутності, читати із БДР.

Складність використання цього підходу полягає у виборі булевої операції, яка буде використовуватися при злитті БДР. Можна розділити коригування на додавання й вилучення значень і створити дві коригувальні БДР: одну для додавання значень з використанням операції АБО та одну для вилучення значень з використанням операції І.

Проте цей підхід не є оптимальним. Якщо порівняти збережені значення із тимчасовими, то можна визначити, як саме змінилося значення. Тоді у коригувальній БДР для кожного зміненого значення необхідно буде встановити значення 1 та злити її із основною БДР з використанням операції «виключне або» (XOR). Завдяки цьому внесення змін можна організувати за одну операцію злиття.

Особливо корисним використання XOR стає, коли необхідно зберігати інформацію не лише про факт наявності дуги між вузлами, а й зберігати вагу цієї дуги. Для цього пропонується виділити додаткову групу змінних, яка відповідатиме за представлення ваг у БДР. Позначимо такі змінні як булевий вектор s . Значення 0 резервуємо для позначення відсутності дуги. Такий підхід не передбачає збереження дробових чисел, проте можна розробити нумерацію дробових чисел і зберігати не саме число, а його номер.

Для того, щоб прочитати значення ваги, під час проходження по БДР, необхідно:

- зафіксувати, які змінні з групи s трапляються на шляху;
- перевірити, чи створює така змінна альтернативний маршрут: якщо так, то необхідно перевірити обидва маршрути та відкинути хибний.

На основі зафіксованих змінних формується значення ваги. Будемо вважати, що кожній парі номерів вузлів може відповідати лише одне значення ваги. У загальному випадку можемо отримати декілька дійсних маршрутів у БДР і, відповідно, декілька значень. Така ситуація може виникнути, якщо замість номерів двох вузлів у запиті буде вказано номер вузла та значення ваги: у цьому разі зможемо отримати номери усіх вузлів, до яких є дуги від вказаного вузла із вказаною вагою. Можливість створення таких запитів вигідно виділяє БДР серед інших способів представлення графів.

Зберігання значень ваг у бінарному вигляді не дає можливості вибирати значення, які більші чи менші за вказане. Проте, якщо планується використовувати невеликі цілі значення ваг, можна перейти від бінарної до унарної системи числення. Це збільшить кількість змінних у БДР, проте мало вплине на її розмір, оскільки більшість нових змінних будуть присутніми в діаграмі лише раз. Завдяки унарній системі числення можна здійснювати вибірки за умовами $\geq$ та $<$. Наприклад, щоб вибрати усі дуги з вагою менше 5, достатньо злити основну БДР із функцією $\neg c_5$. Окрему змінну можна виділити для зберігання знаку.

Цей підхід дає можливість накопичувати зміни і фіксувати їх за необхідності, завдяки чому зменшує кількість операцій злиття БДР. Проте він вимагає значного ускладнення проєкту та додаткових витрат пам'яті для реалізації гарячого сховища.

У розроблюваній системі було виділено три основні сутності: агент, сесія та об'єкт. Агент – це користувач рекомендаційної системи. Сесія – це звернення агента до рекомендаційної системи, в ході якого йому буде сформовано список рекомендацій. Один агент може мати скільки завгодно сесій. Об'єкт – це елемент контенту комп'ютерної мережі, що може бути рекомендований агенту.

Користувач має доступ до чотирьох основних дій у моделі: задання початкових параметрів рекомендаційної системи, створення агентів, створення об'єктів та створення сесій.

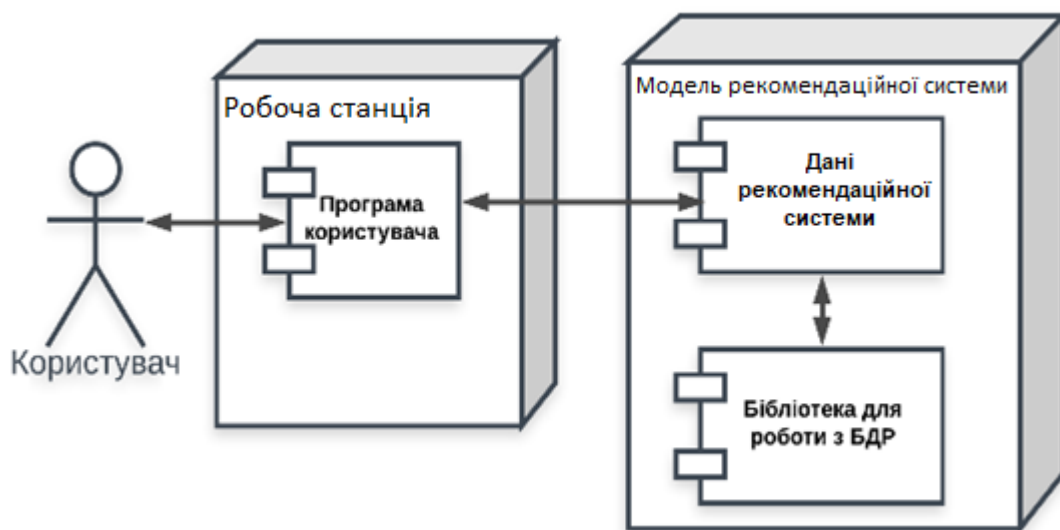


Рис. 3.3. Діаграма розгортання програмної моделі рекомендаційної системи

Запропонована реалізація моделі не призначена для самостійної роботи, тому взаємодію з нею користувач повинен здійснювати через іншу програму, яка використовуватиме інтерфейс моделі (рис. 3.4).

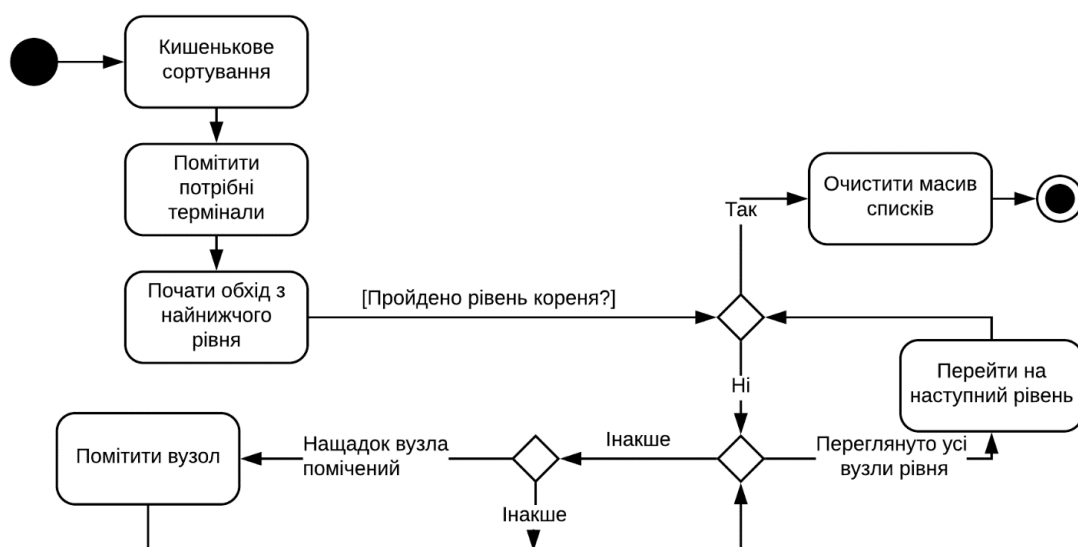


Рис. 3.4. Діаграма діяльності ініціювання пошуку в БДР

На рис. 3.5 показано, як взаємодіють користувацька програма та компоненти моделі рекомендаційної системи під час вибору агентів для формування рекомендацій. Модель має визначити, з якими агентами поточний агент має схожі вподобання, а також які об'єкти можуть йому сподобатися, виходячи з дій схожих на нього агентів. Оскільки дані про агентів та об'єкти зберігаються у різних діаграмах, то для отримання кінцевого результату необхідно об'єднати отримані дані. Запропонована реалізація моделі рекомендаційної системи повинна ефективно використовувати ресурси комп'ютера, оскільки моделювання поведінки великої кількості агентів та об'єктів вимагає значного об'єму пам'яті. Також, оскільки модель має використовуватися як бібліотека, а не як самостійний продукт, то вона повинна легко інтегруватися з іншими програмними продуктами. Для реалізації моделі було обрано мову програмування C. Наш вибір зумовлений наступними чинниками.

1. У мові C реалізовано пряме керування пам'яттю. Це дає можливість мінімізувати супутні витрати пам'яті, але також може стати джерелом таких вразливостей, як витік пам'яті чи переповнення буфера [20].

2. Багато мов програмування, такі як Python, C# та NodeJS, можуть використовувати розширення, написані на C.

3. Код, написаний на C, добре переноситься між платформами та операційними системами.

Для реалізації бінарних діаграм рішень необхідно визначити спосіб представлення БДР у пам'яті та реалізувати операції створення та редукування та редагування БДР, пошуку значення БДР за параметрами та оптимізації порядку змінних. Було вирішено спиратися на ідеї, запропоновані Д. Кнудом у роботі [18], а також ввести деякі власні оптимізації.

Питання ефективного використання пам'яті при роботі з БДР стоїть особливо гостро, оскільки вони мають ефективно представляти булеві функції великої розмірності. Наприклад, у роботі [18, с. 276] показано, що не більше $2^{2^{46}}$ функцій від 98 змінних мають БДР розміром менше трильйона, тоді як

розмір інших має порядок $O(2^{98})$. Тому більшість оптимізацій спрямовано на економію пам'яті. Приклад роботи програмної моделі зображено на рис. 3.5.

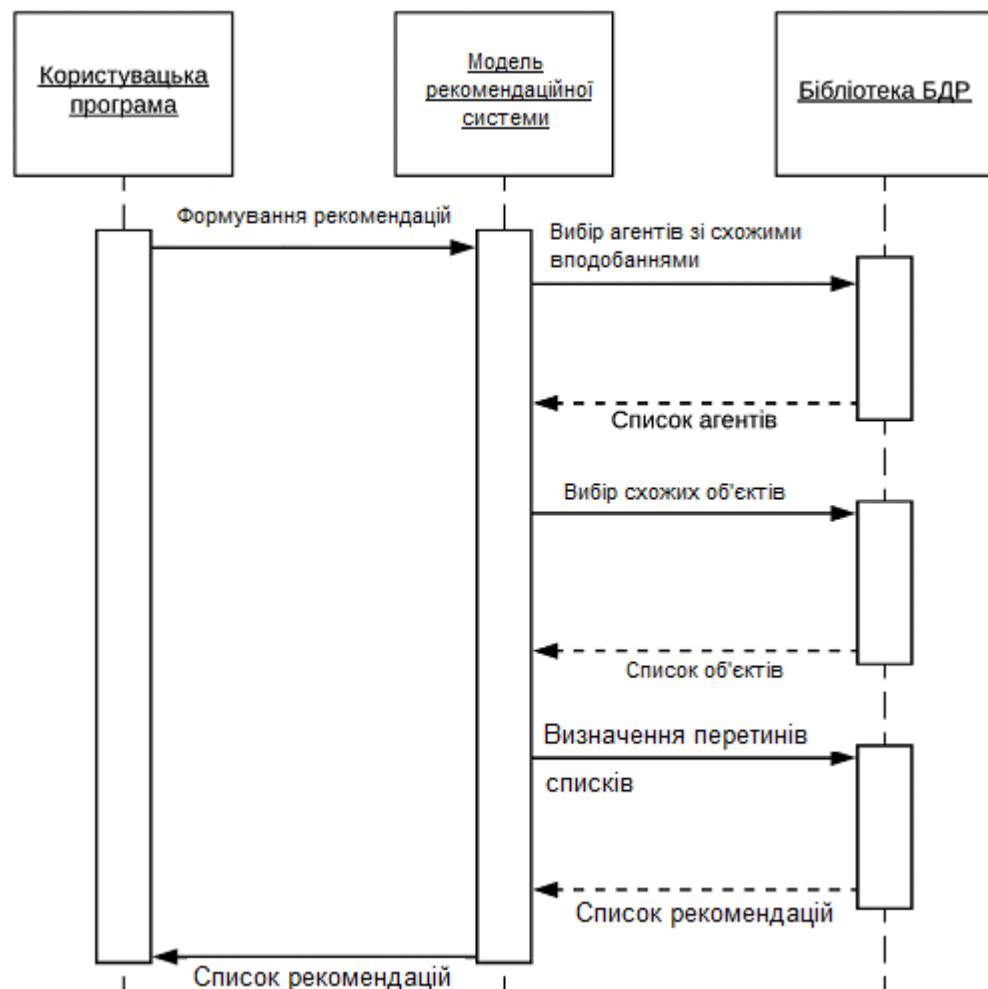


Рис. 3.5. Діаграма діяльності щодо ініціювання пошуку
в бінарних діаграмах рішень

Для прикладу було згенеровано випадковий граф структури соціальної мережі рекомендаційної системи.

Для роботи з моделлю спочатку створюємо нову модель, у якій міститься *agentCount* агентів. Функція *createModel* створює БДР, необхідні для роботи моделі, та змінює їхні розміри відповідно до кількості агентів у мережі. Наприклад, для представлення зв'язків між агентами БДР повинна містити три

групи змінних, кожна з яких повинна мати можливість описати *agentCount* змінних.

Далі ми створюємо масив *varParams*, який необхідний для іменування змінних, збережених у БДР. У нього буде записуватися результат перетворення маршруту всередині БДР на групи змінних та їхні значення.

У циклі для кожного агента випадковим чином обираємо іншого агента та встановлюємо між ними взаємний зв'язок за допомогою двох викликів функції *modelAddContact*.

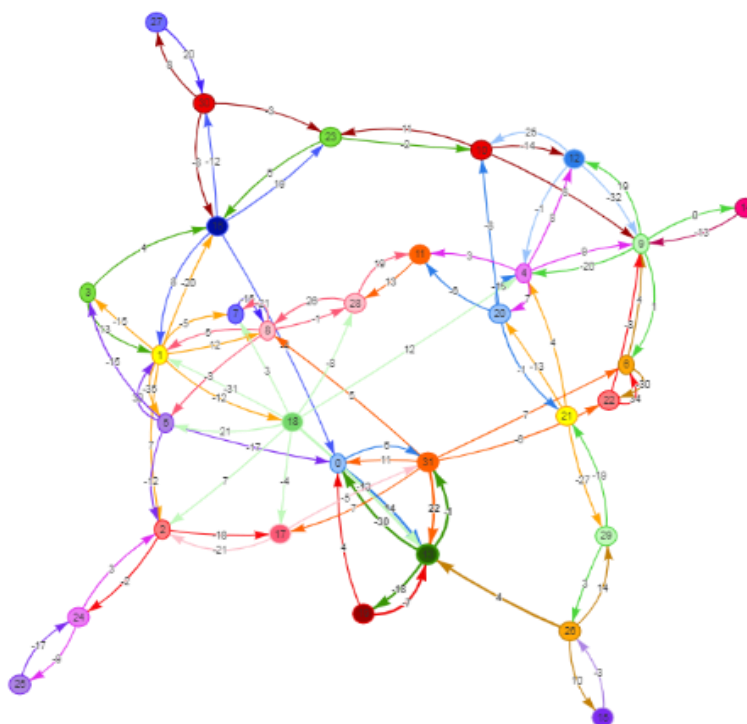


Рис. 3.6. Граф соціальної мережі
рекомендаційної системи з 32 вузлів

Можна побачити, що ця мережа схожа на мережі «тісного світу» та безмасштабні мережі, оскільки багато агентів пов'язані між собою у коло, проте існують також відокремлені групи агентів (рис. 3.6.). Змінивши правила встановлення нового контакту, можна наблизитися до одного із цих видів мереж.

3.3.2. Опис методу програмного імітаційного моделювання рекомендаційної системи

Рекомендаційна система отримує наступні параметри: n_a – кількість агентів, n_s – кількість сесій, n_i – кількість об'єктів, n_{al} – максимальна кількість вподобань агента, n_{sl} – максимальний розмір сесії.

Рекомендаційна система у розробленій програмній моделі працює наступним чином.

Для кожного з n_a агентів випадковим чином генерується від 1 до n_{al} вподобань (рис. 3.7). При цьому унікальність вподобань не перевіряється, тому реальна кількість вподобань може виявитися меншою.

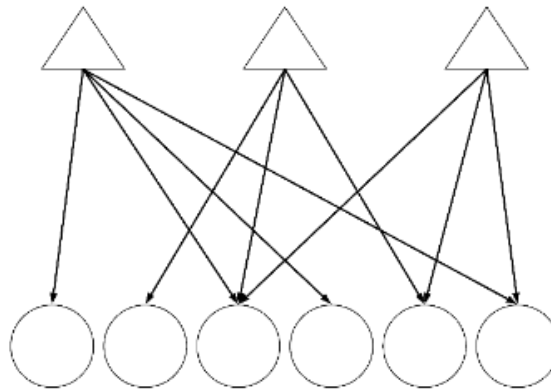


Рис. 3.7. Створення вподобань для агентів
(трикутники – агенти, круги – об'єкти)

Створюється n_s сесій. До кожної сесії закріплюється випадковим чином обраний агент. Потім серед вподобань цього агента випадковим чином обирається від 1 до $\min(n_{al}, n_{sl})$ вподобань, які копіюються до сесії (рис. 3.8).

Випадковим чином обирається контрольна сесія, для якої буде сформовано рекомендацію.

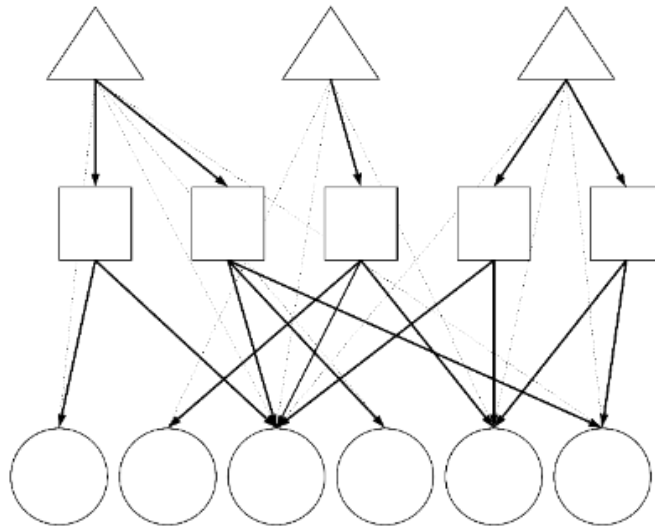


Рис. 3.8. Створення сесій та вподобань для них (трикутники – агенти, квадрати – сесії, круги – об’єкти; сесія містить вподобання агента, які він зробив за одне відвідування ресурсу)

Визначаються усі об’єкти, які належать до контрольної сесії (рис. 3.9).

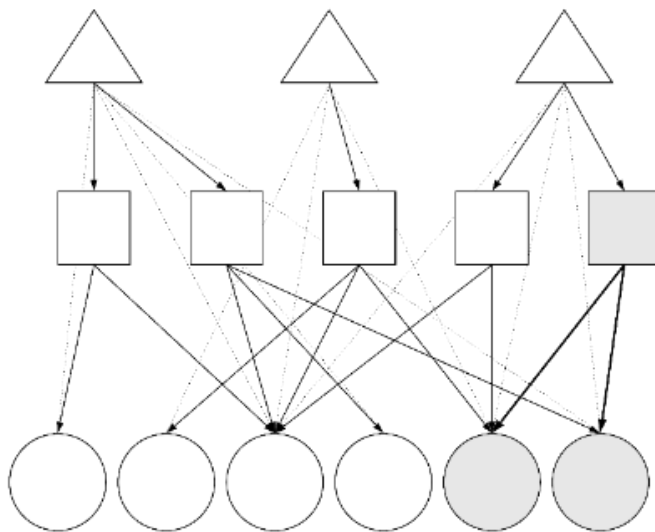


Рис. 3.9. Визначення контрольної сесії
(трикутники – агенти, квадрати – сесії, круги – об’єкти)

Здійснюється пошук усіх сесій, вподобання яких мають перетин із вподобаннями контрольної сесії (рис. 3.10). На цьому етапі є можливість відфільтрувати сесії за розміром перетину.

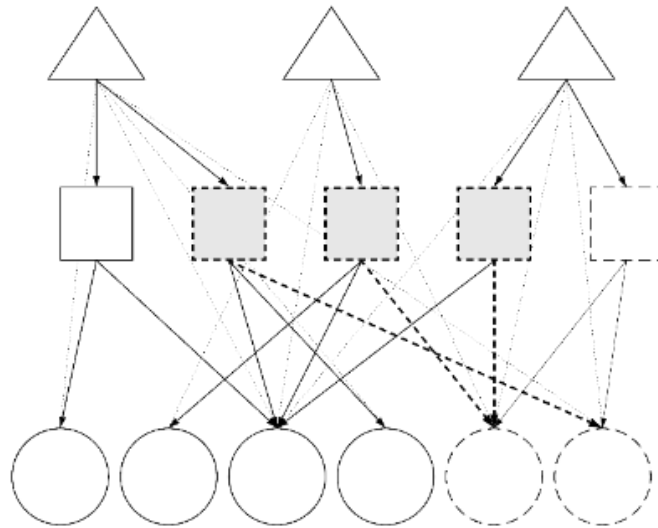


Рис. 3.10. Пошук сесій, які мають перетин із контрольною сесією
(трикутники – агенти, квадрати – сесії, круги – об’єкти)

Визначаються об’єкти, які буде рекомендовано. Здійснюється пошук усіх об’єктів, які належать хоча б одній із відібраних сесій, але не належать до контрольної сесії (рис. 3.11). На цьому етапі є можливість відфільтрувати об’єкти за кількістю закріплених сесій.

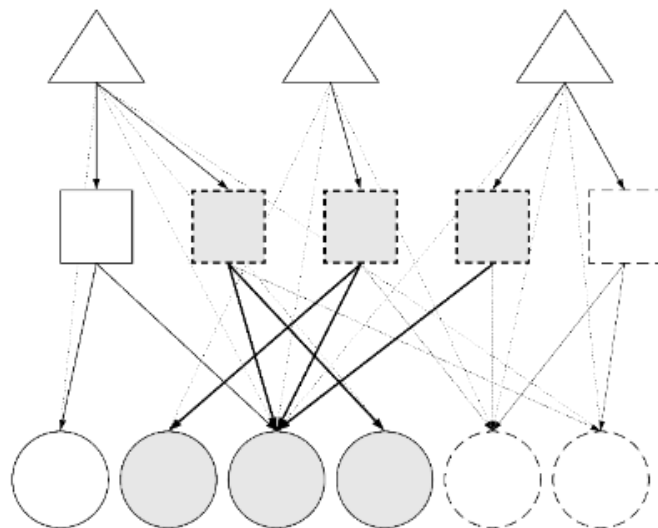


Рис. 3.11. Вибірка об’єктів для формування рекомендації
(трикутники – агенти, квадрати – сесії, круги – об’єкти)

Таким чином, етапи роботи запропонованого методу моделювання рекомендаційної системи полягають у наступному:

Етап 1. Ініціалізація рекомендаційної системи: задається кількість агентів, предметів та сесій, а також розмір сесії та максимальна кількість вподобань.

Етап 2. Для кожного агента випадковим чином генерується від 1 до n_{al} вподобань.

Етап 3. Створюється n_s сесій. До кожної сесії закріплюється випадковим чином обраний агент. Потім серед вподобань цього агента випадковим чином обирається від 1 до k вподобань, які копіюються до сесії.

Етап 4. Випадковим чином обирається контрольна сесія, для якої буде сформовано рекомендацію.

Етап 5. Визначаються усі предмети, які належать до контрольної сесії.

Етап 6. Здійснюється пошук усіх сесій, вподобання яких мають перетин із вподобаннями контрольної сесії. На цьому етапі є можливість відфільтрувати сесії за розміром перетину.

Етап 7. Визначаються предмети, які буде рекомендовано. Здійснюється пошук усіх предметів, які належать хоча б одній із відібраних сесій, але не належать до контрольної сесії. На цьому етапі є можливість відфільтрувати предмети за кількістю закріплених сесій.

3.2. Розробка методу зберігання даних рекомендаційної системи на основі бінарних діаграм рішень

На сьогоднішній день рекомендаційні системи мають широке застосування у соціальних мережах, системах Інтернет-торгівлі, поширенні медіа-контенту, реклами тощо [22-25]. Ефективний спосіб представлення даних, необхідних для роботи такої системи, може зменшити кількість потрібних ресурсів та полегшити розробку і використання більш складних алгоритмів для формування списків рекомендацій. Спосіб збереження даних

рекомендаційної системи є важливим з точки зору якості її роботи, швидкості, можливості масштабування, зручності виконання основних операцій із даними для формування рекомендацій.

Все частіше для зберігання даних рекомендаційних систем та інших додатків починають використовувати графові моделі [26-31]; також графова форма представлення даних стає поширеною у програмному моделюванні складних систем та мереж [26, 27], і це відбувається через ряд переваг графових моделей. Такий спосіб зберігання даних надає не лише зручний формат зберігання даних, а й зручний формат запитів до бази даних.

Існують різні способи представлення графів в комп'ютерних системах, зокрема таблиці суміжності, списки ребер, матриці інцидентності тощо [32]. Також графи можна зберігати у вигляді бінарних діаграм рішень, які по суті є економною формою представлення булевих функцій у вигляді орієнтованого ациклічного графа [18-20].

На даний момент практично не досліджено можливість використовувати бінарні діаграми рішень для зберігання даних рекомендаційної системи.

Було розроблено метод зберігання даних рекомендаційної системи у формі бінарних діаграм рішень, а також проведено дослідження ефективності даного методу шляхом експериментів на запропонованій програмній моделі рекомендаційної системи.

Проведено серію експериментів для перевірки ефективності запропонованого способу зберігання даних на основі бінарних діаграм рішень, та порівняння його з іншими способами зберігання даних в інших структурах даних.

У проведених експериментах було здійснено порівняння чотирьох різних реалізацій – на основі бінарних діаграм рішень, бітових масивів, зв'язних списків та хеш-таблиць. Кожна реалізація дотримується описаного вище алгоритму програмного моделювання рекомендаційної системи, але має певні відмінності.

Реалізація на основі бітових масивів найпростіша, оскільки у цьому випадку є можливість отримати прямий доступ до значення за номером агента/сесії та об'єкту, але це потребує значних витрат пам'яті.

Інші структури даних більш складні, тому для прискорення їх роботи для збереження інформації про перетин із контрольною сесією було застосовано хеш-таблиці. Також вони не дають можливості вибирати випадковий елемент, тому на етапі створення сесій відбувається повний перебір вподобань агента, доки вони не закінчаться або доки не буде відібрано необхідну кількість вподобань. Для кожного вподобання агента обирається випадкове число, на основі якого визначається, чи заносити вподобання до сесії.

Зв'язні списки дозволяють зберігати інформацію лише про існуючі вподобання, завдяки чому вони використовують менше пам'яті, ніж бітові масиви. Проте через те, що вони не дають доступу до елементів у випадковому порядку, пошук перетину двох сесій займає багато часу.

Хеш-таблиці дозволяють отримати доступ до елементу за ключем, завдяки чому можна покращити результати, отримані за допомогою зв'язних списків. Якщо у випадку зв'язних списків складність пошуку перетину квадратична, то у випадку хеш-таблиці вона лінійна.

Дані рекомендаційної системи під час експериментів зберігалися в оперативній пам'яті з можливістю зберігання у текстових файлах. Для експериментів використовувався персональний комп'ютер з 24 Гб оперативної пам'яті. Прочерки у комірках показують, що неможливо було отримати результат через нестачу оперативної пам'яті.

Результати експериментів наведені у табл. 3.1, де:

- agentCount – кількість агентів;
- itemsCount – кількість об'єктів;
- sessionsCount – кількість сесій;
- sessionsSize – розмір сесії;
- maxLikes – максимальна кількість лайків.

Таблиця 3.1. Результати експериментів для дослідження ефективності використання бінарних діаграм рішень у програмній моделі рекомендаційної системи

Структура даних	Використана пам'ять, мін. знач., Gb	Використана пам'ять, макс. знач., Gb	Час генерації лайків, ms	Час генерації сесій, ms	Час генерації рекомендацій, ms
<i>agentCount = 4096, itemsCount = 8192, sessionsCount = 16384, sessionSize = 64, maxLikes = 640</i>					
БДР	0,024	0,037	2562	2209	411
Хеш-таблиця	0,106	0,106	76	103	25
Бітовий масив	0,166	0,166	12	127	111
Зв'язний список	0,108	0,108	78	120	247
<i>agentCount = 16384, itemsCount = 32768, sessionsCount = 65536, sessionSize = 128, maxLikes = 1024</i>					
БДР	0,17	0,29	31408	34551	5240
Хеш-таблиця	0,46	0,46	369	702	179
Бітовий масив	2,5	2,5	179	1327	1623
Зв'язний список	0,79	0,79	516	721	2823
<i>agentCount = 32768, itemsCount = 65536, sessionsCount = 131072, sessionSize = 192, maxLikes = 1536</i>					
БДР	0,452	0,733	95075	106675	23327
Хеш-таблиця	1,1	1,1	1168	2013	640
Бітовий масив	-	-	-	-	-
Зв'язний список	2,2	2,2	1499	2112	9973
<i>agentCount = 65536, itemsCount = 131072, sessionsCount = 262144, sessionSize = 192, maxLikes = 1536</i>					
БДР	0,786	1,3	210945	237214	45766
Хеш-таблиця	2,2	2,2	2312	4034	316
Бітовий масив	-	-	-	-	-
Зв'язний список	4,5	4,5	2989	4323	23223
<i>agentCount = 131072, itemsCount = 262144, sessionsCount = 524288, sessionSize = 256, maxLikes = 2048</i>					
БДР	1,8	3,6	607610	682754	153045
Хеш-таблиця	6,4	6,4	6082	12327	3954
Бітовий масив	-	-	-	-	-
Зв'язний список	12,1	12,1	8052	11195	74346
<i>agentCount = 262144, itemsCount = 524288, sessionsCount = 1048576, sessionSize = 256, maxLikes = 2048</i>					
БДР	3,4	7,2	1366467	1551391	313122
Хеш-таблиця	12,8	12,8	12268	24666	6221
Бітовий масив	-	-	-	-	-
Зв'язний список	24	24	16051	22323	146845
<i>agentCount = 524288, itemsCount = 1048576, sessionsCount = 2097152, sessionSize = 256, maxLikes = 2048</i>					
БДР	6,5	14,4	3170877	3623865	643864
Хеш-таблиця	25,5	25,5	24204	116728	44527
Бітовий масив	-	-	-	-	-
Зв'язний список	-	-	-	-	-

Бінарні діаграми рішень мають як переваги, так і недоліки у порівнянні з хеш-таблицями, бітовими масивами та зв'язними списками. БДР дають можливість шукати дані за частковими ключами, що додатково дозволяє зберігати зв'язки більшої розмірності. Бінарні діаграми даних зберігають дані з меншими витратами пам'яті, ніж хеш-таблиці, завдяки чому у оперативній пам'яті одночасно можна розмістити інформацію про більшу кількість вподобань. БДР програють у швидкодії, проте було застосовано кілька оптимізацій:

- підтримка унікальності фрагментів БДР обчислювально дуже дорога, тому вона була відділена від процедури редагування і виконується кілька разів за час генерації тестових даних;

- видалення надлишкових фрагментів з БДР спричиняє розрідженість вузлів у пам'яті, через що зростає час доступу до вузла. Тому після видалення надлишкових вузлів вони ущільнюються шляхом перенесення у нову область пам'яті;

- ущільнення вузлів дає можливість звільнити певну кількість пам'яті, проте виявилось, що звільнення пам'яті у цій ситуації неефективне – наступні операції редагування вимагають створення нових вузлів, через що потрібно буде повторно переносити вузли між областями пам'яті;

- для здійснення пошуку у БДР її необхідно проіндексувати відповідно до ключа та параметрів пошуку, а обхід усіх вузлів БДР – це дорога операція. Для прискорення пошуку було покращено процедуру індексування, завдяки чому пошук із вказаною першою частиною ключа не вимагає повторної індексації.

Результати експерименту показали, що у випадку застосування бінарних діаграм рішень оперативна пам'ять використовується більш економно в порівнянні із хеш-таблицями, бітовими масивами та зв'язними списками. Для компенсації меншої швидкості роботи з даними в оперативній пам'яті при використанні БДР запропоновано кілька оптимізацій. Завдяки меншому

використанню оперативної пам'яті можна зберігати інформацію про більшу кількість вподобань користувачів рекомендаційної системи при виконанні обчислень для формування рекомендацій. Зазначений факт є суттєвим у випадку великих розмірів графів, що характерно для реальних рекомендаційних систем. Крім того, можливість пошуку даних за частковими ключами в бінарних діаграмах рішень дозволяє додатково збільшити розмірність даних, що зберігаються.

3.3. Проведення експериментів для дослідження різних стратегій редагування бінарних діаграм рішень у роботі з графовими даними рекомендаційної системи

Можна вважати, що БДР отримується із бінарного дерева рішень шляхом вилучення усіх надлишкових підструктур (ізоморфних підграфів). За допомогою БДР можна зберігати лише булеві та псевдобулеві функції, а також дані, що можуть бути приведені до таких даних, наприклад граф соціальної мережі. БДР дають можливість зберігати дані у стисненому вигляді та швидко отримувати значення функції за її параметрами, але редагування БДР вимагає складних обчислень [18].

Для занесення нового значення до БДР або зміни існуючого, її необхідно відредагувати. Було використано наступний алгоритм редагування БДР:

1. Створити термінал з новим значенням, якщо він ще не існує.
2. Здійснити спуск від кореня БДР до терміналу відповідно до значень параметрів.
3. Визначити перший рівень l , на якому вузол належить кільком різним значенням. Такий вузол має кілька посилок з верхніх рівнів або відсутній у рівні (відповідно до правил побудови БДР це означає, що цей вузол мав однакових нащадків). Якщо такого рівня немає – направити останній вузол на новий термінал і припинити роботу.

4. Побудувати нову гілку, яка за вказаними параметрами вказуватиме на новий термінал. Для підтримки редукованості БДР ми маємо використовувати вузли повторно. Для рівнів від l і вище гарантується, що вузол не використовується у іншій гілці, тож за необхідності його можна змінити або видалити.

Рівні БДР утворені за допомогою однозв'язних списків. Кожен вузол має посилання на свого сусіда у рівні. Один рівень об'єднує усі вузли, які відносяться до однієї булевої змінної. Під час тестів на великих об'ємах даних було виявлено, що цей підхід вимагає дуже багато обчислювальних ресурсів, оскільки для пошуку вузлів для повторного використання або видалення вузлів із однаковими нащадками доводиться здійснювати повний обхід рівня.

Перша застосована оптимізація полягала у винесенні підтримки редукованості із процедури редагування. Це спричиняє більші витрати пам'яті через створення надлишкових вузлів, проте дозволяє рідше виконувати складну процедуру підтримки редукованості.

Перший алгоритм, який було застосовано для підтримки редукованості – алгоритм, запропонований Д. Кнутом у роботі [18]. Він дає можливість видалити зайві вузли за один прохід без застосування додаткової пам'яті. Але для роботи цього алгоритму потрібні поля, які ми використовуємо для зв'язку вузлів у рівні, тому після редукування БДР потрібно здійснити ще один обхід для відновлення списків. Подальше тестування показало, що редукування на місці сповільнює роботу редагування. Тому після редукування було додано ще один крок, який полягає у наступному:

1. Виділяється блок пам'яті, еквівалентний блоку, виділеному для вузлів БДР.
2. БДР відтворюється у новому блоці рівень за рівнем.
3. Старий блок пам'яті видаляється і замінюється новим.

Таким чином вузли групуються у пам'яті за рівнем, завдяки чому збільшується швидкість оперування даними.

Наступна оптимізація полягає у покращенні кроку компресії вузлів. Оскільки компресія вимагає повного обходу усіх вузлів за рівнями, цей обхід можна використати і для пошуку надлишкових вузлів. При копіюванні вузла у нову область пам'яті до старого вузла заноситься його позиція у новому блоці пам'яті. Якщо нові позиції дочірніх вузлів збігаються – вузол не копіюється, а його нова позиція позначається як нова позиція дочірнього вузла. Для пошуку дубльованих вузлів використовується хеш-таблиця – значенням виступає позиція вузла, а ключем – хеш від позицій дочірніх вузлів. Таким чином, якщо вузол із такими дочірніми вузлами уже занесений до хеш-таблиці, то його не потрібно заносити до нової області пам'яті знову. При цьому підході уже немає необхідності у окремому кроці редукування, що дає можливість відмовитися від двох повних обходів БДР, але вимагає додаткової пам'яті для зберігання хеш-таблиці для одного рівня БДР.

Такий підхід до пошуку надлишкових вузлів дозволив здійснити наступну оптимізацію. Оскільки хеш-таблиця дає можливість знайти вузол за позиціями дочірніх вузлів за сталий час, немає необхідності у підтримці зв'язку між вузлами у одному рівні, тому і видалення вузла можна здійснити за сталий час. У нашій реалізації вузол хеш-таблиці займає 8 байтів, що вдвічі менше, ніж вузол БДР. При цьому, для підтримки нормальної швидкодії хеш-таблиці, вона має бути заповнена не більш ніж на 50%. Таким чином, хеш-таблиця займатиме стільки ж місця, як і БДР. Проте цей підхід позбавляє необхідності здійснювати компресію вузлів, тому хеш-таблиця буде зберігатися на місці, яке займала б копія БДР.

У табл. 3.2 наведено порівняння швидкодії різних стратегій редагування БДР. Як бачимо, стратегія хешування дала найкращі результати. Стратегія редукування і компресії працює за найдовший час, оскільки потребує найбільшої кількості обходів вузлів БДР.

Таблиця 3.2. Порівняння часу, витраченого для заповнення бінарних діаграм рішень при різних стратегіях редагування

№	Кількість записів	Стратегія редукування і компресії, мс	Стратегія редукування під час компресії, мс	Стратегія хешування, мс
1	2621440	215	1024	670
2	16777216	17836	10872	6446
3	50331648	66188	34761	20686
4	100663296	147702	74708	40344

Таким чином було реалізовано та порівняно декілька стратегій редагування бінарних діаграм рішень. Показано, що стратегія хешування вузлів БДР має найвищу швидкість роботи, проте вимагає більшого об'єму пам'яті на постійній основі, а стратегія редукування і компресії БДР має найменшу швидкодію.

3.4. Проведення експериментів для порівняльного аналізу методів зберігання інформації рекомендаційної системи децентралізованої однорангової комп'ютерної мережі на основі різних структур даних

Було проведено серію експериментів для порівняння ефективності використання різних структур даних для зберігання інформації рекомендаційної системи за затратами часу та пам'яті. Результати експериментів наведені у табл. 3.3 – 3.5 та на рисунках 3.12 – 3.13.

Експерименти проводилися на комп'ютері з процесором AMD Ryzen 5 3600 та 32 Гб оперативної пам'яті. Для формування рекомендацій було використано колаборативну фільтрацію. Експерименти проводилися на запропонованій програмній імітаційній моделі рекомендаційної системи. Результати експериментів для порівняння ефективності різних структур даних для зберігання інформації рекомендаційної системи наведено у табл. 3.3, де:

- agentCount – кількість агентів;
- itemsCount – кількість об’єктів;
- sessionsCount – кількість сесій;
- sessionsSize – розмір сесії;
- maxLikes – максимальна кількість лайків.

Таблиця 3.3. Результати експериментів для порівняння ефективності різних структур даних для зберігання інформації рекомендаційної системи

Структура даних	Використана пам’ять, мін. знач., Gb	Використана пам’ять, макс. знач., Gb	Час обробки лайків, ms	Час ініціалізації сесій, ms	Час генерації рекомендацій, ms
<i>agentCount = 65536, itemsCount = 131072, sessionsCount = 262144, sessionSize = 192, maxLikes = 1536</i>					
БДР	0,786	1,3	210945	237214	45766
Хеш-таблиця	2,2	2,2	2312	4034	316
В+-дерево	2,1	2,1	5407	5324	213
Зв’язний список	4,5	4,5	2989	4323	23223
Розгорнутий зв’язний список	0,578	0,578	962	2259	141
Інвертований зв’язний список	0,767	0,767	947	3842	93
<i>agentCount = 131072, itemsCount = 262144, sessionsCount = 524288, sessionSize = 256, maxLikes = 2048</i>					
БДР	1,8	3,6	607610	682754	153045
Хеш-таблиця	6,4	6,4	5359	10811	2312
В+-дерево	5,6	5,6	14925	14554	720
Зв’язний список	12,1	12,1	8052	11195	74346
Розгорнутий зв’язний список	1,2	1,2	2623	5921	412
Інвертований зв’язний список	1,9	1,9	2616	10457	277
<i>agentCount = 262144, itemsCount = 524288, sessionsCount = 1048576, sessionSize = 256, maxLikes = 2048</i>					
БДР	3,4	7,2	1366467	1551391	313122
Хеш-таблиця	12,8	12,8	10661	21722	4248
В+-дерево	11,2	11,2	29810	29111	1357
Зв’язний список	24	24	16051	22323	146845
Розгорнутий зв’язний список	2,4	2,4	5227	11846	808
Інвертований зв’язний список	3,9	3,9	5248	20919	563

Як видно з табл. 3.3, найкращі результати показали методи зберігання з використанням розгорнутого та інвертованого розгорнутого зв'язних списків. Тому було вирішено також провести додаткову серію експериментів з цими структурами даних для різного розміру блоку (кількість елементів в блоці зв'язного списку), результати якої наведені в табл. 3.4.

У табл. 3.4 було використано наступні скорочення:

- РЗС – розгорнутий зв'язний список,
- ІРЗС – інвертований розгорнутий зв'язний список.

Таблиця 3.4. Результати експериментів для порівняння ефективності звичайного та інвертованого розгорнутих зв'язних списків для зберігання інформації рекомендаційної системи

№ експ.	Структура даних	Кількість елементів у блоці зв'язного списку	Використання пам'яті, макс. знач., Gb	Час обробки лайків, ms	Час ініціалізації сесій, ms	Час генерації рекомендацій, ms
1	РЗС	52	4,7	10954	25355	1998
	ІРЗС		7,1	10845	37742	1563
2	РЗС	130	4,4	10577	24480	1754
	ІРЗС		6,9	10393	42919	1281
3	РЗС	200	5,5	10433	24427	1821
	ІРЗС		8,2	10429	42846	1334
4	РЗС	260	4,8	10404	23710	1602
	ІРЗС		7,8	10933	41616	855

Розгорнутий список показав кращі результати за використовуваною пам'яттю в середньому в 1,54 разів та за часом генерації сесій – в середньому в 1,68 разів. В той же час інвертований розгорнутий список показав перевагу за часом генерації рекомендацій в середньому в 1,43 разів. Час генерації лайків обидва методи показали приблизно однаковим.

У табл. 3.5. наведено результати серії експериментів, проведених для порівняння часу формування рекомендацій системою при використанні різних структур даних.

Таблиця 3.5. Результати серії експериментів для порівняння часу формування рекомендацій системою при використанні різних структур даних

№	BDD, мс	Hash map, мс	Linked list, мс	B+ tree, мс	Unrolled linked list, мс	Unrolled linked list + inverted, мс
1	45766	316	23223	213	141	93
2	153045	2312	74346	720	412	277
3	313122	4248	146845	1357	808	563
4	643864	9153	-	2797	1602	855

Примітка. Прочерк у комірці таблиці показує, що було неможливо отримати результат через нестачу оперативної пам'яті.

На основі одержаних результатів із табл. 3.5 було побудовано графік, наведений на рис. 3.12.

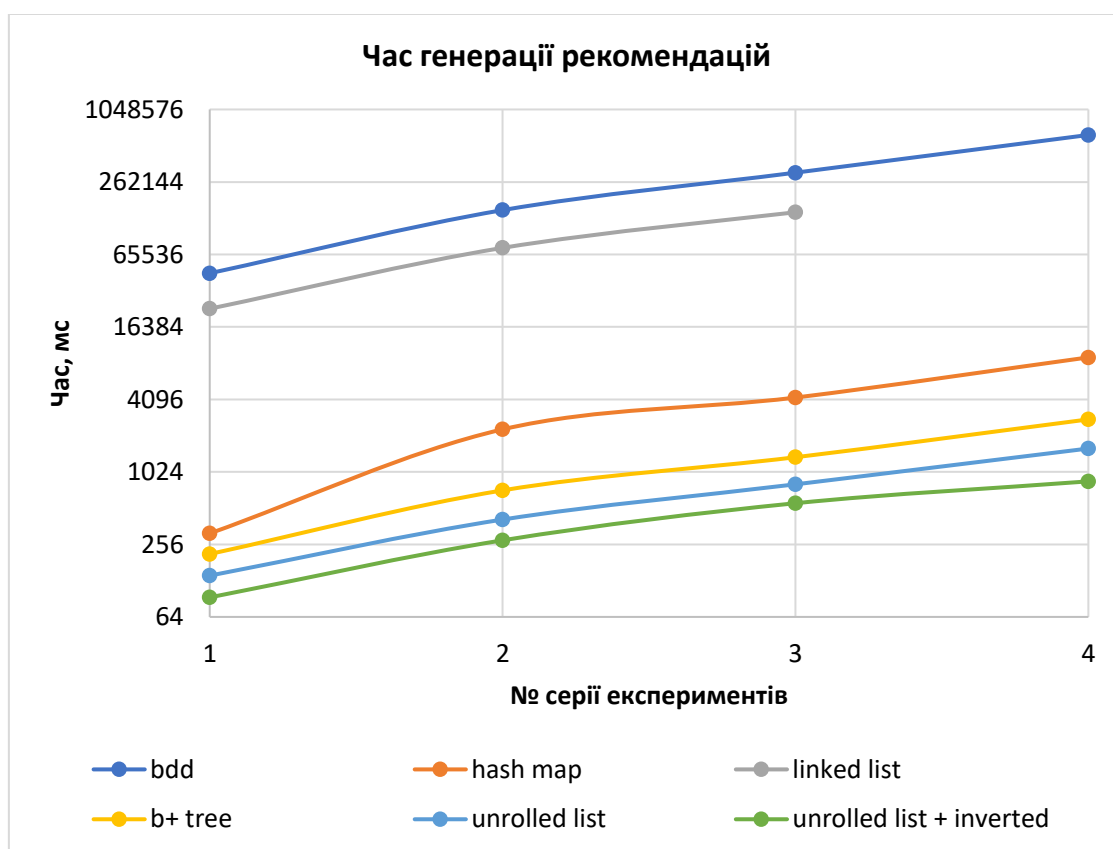


Рис. 3.12. Час генерації рекомендацій системою для кожної з розглянутих структур даних

Як видно з таблиці та рисунка, найкращі результати по часу формування рекомендацій показали наступні структури даних: звичайний розгорнутий список та інвертований розгорнутий список. Також непогані результати показала структура даних В+-дерево.

У таблиці 3.6 наведено результати серії експериментів, проведених для порівняння кількості використаної пам'яті у процесі формування рекомендацій системою при використанні різних структур даних для реалізації бази даних.

Таблиця 3.6. Результати серії експериментів для порівняння кількості використаної пам'яті у процесі формування рекомендацій системою при використанні різних структур даних

№	BDD, Гб	Hash map, Гб	Linked list, Гб	B+ tree, Гб	Unrolled linked list, Гб	Unrolled linked list+ inverted, Гб
1	1,3	2,2	4,5	2,1	0,578	0,767
2	3,6	6,4	12,1	5,6	1,2	1,9
3	7,2	12,8	24	11,2	2,4	3,9
4	14,4	25,5	-	22,3	4,8	7,8

Примітка. Прочерк у комірці таблиці показує, що було неможливо отримати результат через нестачу оперативної пам'яті.

На основі одержаних результатів з табл. 3.6 було побудовано графік, наведений на рис. 3.13. Як видно з таблиці та рисунка, найкращі результати за використаною пам'яттю для формування рекомендацій, аналогічно як і з результатами за затратами часу, показали наступні структури даних: звичайний розгорнутий список, інвертований розгорнутий список та В+-дерево.

Профілювання тестового коду показало, що значна відмінність у часі генерації сесій у варіантів з та без інвертованого списку спричинена затримкою доступу до елементів інвертованого списку, оскільки доступ постійно здійснюється до різних елементів. З реальними даними рішення з розгорнутим списком працюватиме дещо повільніше, оскільки кешування буде менш ефективним.

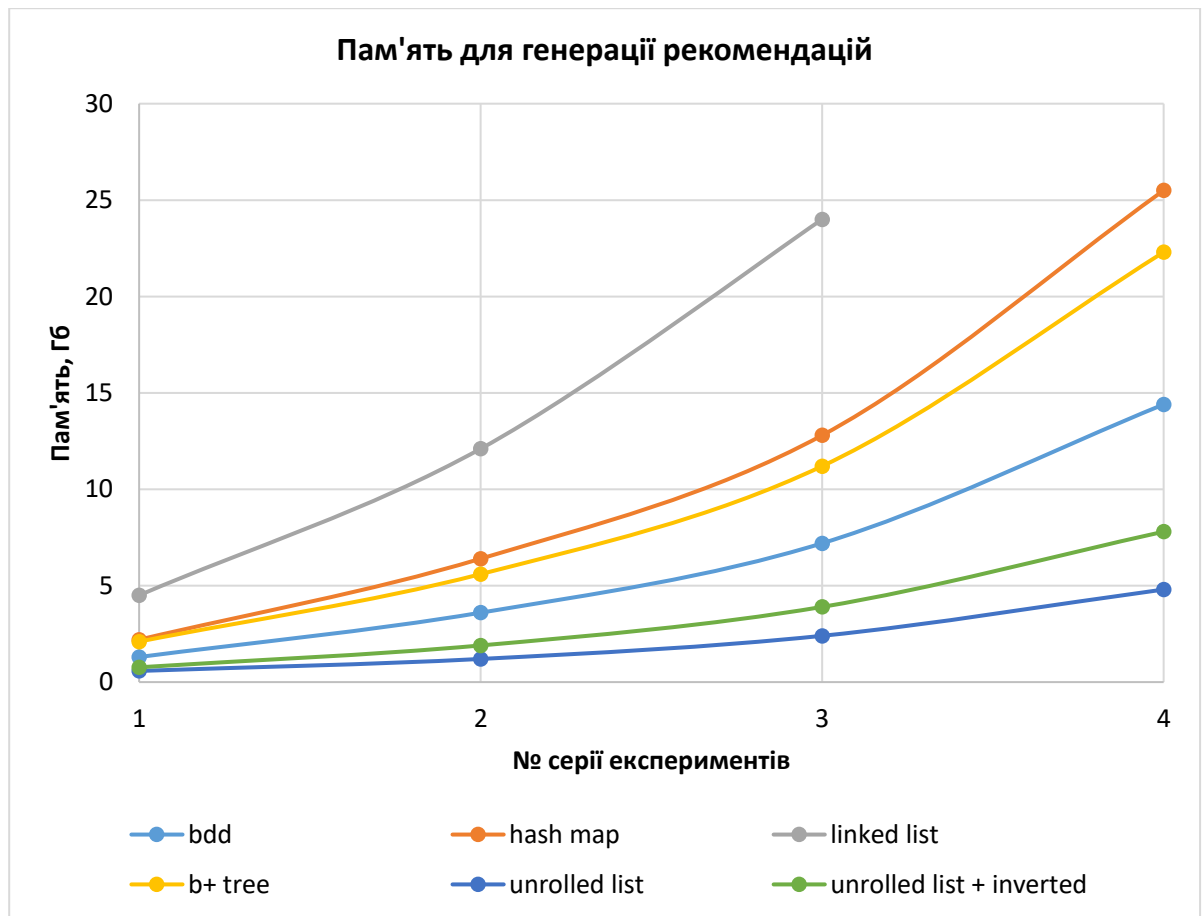


Рис. 3.13. Використана пам'ять для генерації рекомендацій системою для кожної з розглянутих структур даних

Структура розгорнутого списку дуже проста, тож в подальшому це дасть можливість використовувати багатопотокову роботу без блокувань.

Структура B+ tree показала результати, близькі до хеш-таблиці. Час доступу до окремого елементу в обох випадках сталий, але B+ tree має певні

переваги: елементи зберігаються відсортованими, а при зміні розміру немає необхідності розширювати область пам'яті.

Розмір блоку розгорнутого списку впливає на швидкість роботи і на об'єм використаної пам'яті. Зменшення розміру блоку дозволяє зменшити втрати пам'яті, але збільшує час доступу до елементів.

Для прискорення пошуку окремих елементів у розгорнутому списку після заповнення блоку можна відсортувати його елементи. Це дасть можливість використовувати бінарний пошук замість лінійного та перевіряти лише ті блоки, де шуканий елемент належить до інтервалу, утвореного найменшим та найбільшим елементами блоку.

Перевага розгорнутого списку над іншими розглянутими структурами даних у використанні пам'яті значною мірою за рахунок того, що зберігається лише факт вподобання, без параметрів. При використанні 4 байт на елемент для зберігання параметра витрати пам'яті наближаються до максимальних витрат бінарних діаграм рішень.

У проведених експериментах розгорнутий список показав найкращі показники швидкодії та використання пам'яті. Профілювання показало, що 75% часу роботи тесту для варіанта з розгорнутим списком зайняло генерування випадкових даних для тесту, тож саме сховище даних має високі показники ефективності. Профілювання варіанта з інвертованим списком показало, що доступ до випадкових блоків займає більше часу через неможливість закешувати їх, тож за умов реального навантаження час вставки нових даних буде більшим, а відносна ефективність застосування інвертованого списку зросте.

Для найбільш ефективного використання пам'яті розмір блоку зв'язного списку має бути адаптований таким чином, щоб блоки були максимально заповнені. Блоки малого розміру зменшують втрати пам'яті, але збільшують час обходу усіх елементів списку та збільшують накладні витрати пам'яті.

3.5. Розробка методу зберігання даних рекомендаційної системи на основі розгорнутих зв'язних списків та хеш-таблиць

На основі проведених досліджень було вирішено для зберігання даних рекомендаційної системи використати розгорнуті лінійні списки та хеш-таблиці, оскільки розгорнуті лінійні списки показали найкращі результати у проведених експериментах, а хеш-таблиці якнайкраще адаптуються до структури однорангових децентралізованих мереж та показали середні результати в тестах.

Також у розробленому методі зберігання даних рекомендаційної системи використаний метод Delta-encoding.

Delta-encoding – це метод, який полягає у представленні ряду чисел як ряду різниць між сусідніми числами, тобто замість ряду $n_0, n_1, n_2, \dots, n_j, \dots$ зберігається ряд $n_0, n_1 - n_0, n_2 - n_1, \dots, n_j - n_{j-1}, \dots$ [33].

Якщо перед застосуванням цього методу відсортувати числа за зростанням, то усі елементи нового ряду гарантовано будуть додатними та меншими, ніж члени початкового ряду. Це відкриває можливості для оптимізації використання пам'яті, оскільки новим числам потрібно менше бітів для представлення у пам'яті. Головний недолік цього підходу полягає у тому, що між елементами є рекурсивна залежність, і для доступу до випадкового елемента необхідно вирахувати значення усіх його попередників.

Для оптимізації методом delta-encoding було обрано структуру даних «розгорнутий зв'язний список». Це зв'язний список, кожен елемент якого містить масив логічних елементів. Блокова структура дозволяє запровадити додаткову оптимізацію: кожен елемент списку містить окрему послідовність чисел, оброблених за допомогою delta-encoding, а перше число зберігається розпакованим. Завдяки цьому можна швидко визначити блок, у якому має знаходитися шукане значення.

Для того, щоб використовувати менше пам'яті, необхідно виділяти для представлення різниці між числами менше байтів, ніж на початкові числа. У цьому випадку необхідно мати алгоритм розв'язання ситуації, коли для представлення різниці необхідно більше байтів, аніж було передбачено.

Одним з варіантів вирішення цієї проблеми є VarLen Encoding [34]. Цей метод передбачає можливість виділяти для числа лише необхідну кількість байтів, використовуючи старший біт кожного байту як маркер кінця послідовності. Головний недолік цього методу полягає у складній процедурі кодування та декодування даних.

Один із запропонованих нами альтернативних методів схожий із VarLen Encoding. Значення різниці розбивається на комірки таким чином, що якщо воно більше, ніж максимальне допустиме значення комірки, то у комірку записується максимальне значення, а у наступну комірку записується зменшена різниця. Таким чином, при використанні комірок по два байти для представлення різниці у 1000000 необхідно $\lceil 1000000 / 2^{16} \rceil = 16$ комірок, або 32 байти. Цей метод простіший у кодуванні/декодуванні, але вимагає більших витрат пам'яті.

Наступний запропонований нами метод полягає у тому, щоб виділяти достатньо байтів для розміщення числа повністю, зарезервувавши перший біт, якщо різниця перевищує максимальне значення комірки. Це дозволяє скоротити накладні витрати і полегшити відтворення чисел. Головний недолік цього підходу у тому, що для додавання числа всередину послідовності блок необхідно створити знову.

Наступний запропонований нами метод використовує особливості структури розгорнутого списку. У випадку, коли вставка відбувається у кінець блоку, а різниця перевищує максимальне значення комірки, то перевіряється можливість вставки на початок наступного блоку. Якщо це неможливо, то для числа створюється новий блок. Цей варіант усуває необхідність зберігати у блоці гетерогенні дані, проте спричиняє підвищені витрати пам'яті за рахунок

часткового заповнення блоків.

Було запропоновано кілька варіантів застосування методу delta-encoding для представлення даних рекомендаційної системи. Кожен із них має свої переваги та недоліки, тому вибір необхідно робити на основі експерименту над типовими даними системи.

При зберіганні даних рекомендаційної системи одними з важливих параметрів бази даних є швидкість операцій читання/запису інформації, а також необхідний об'єм пам'яті для збереження даних у тому чи іншому форматі. Тому для рекомендаційних систем часто доцільно використовувати прості моделі даних.

Пропонується зберігати у розгорнутому списку рекомендації користувачам для кожної сесії роботи рекомендаційної системи. А у варіанті із інвертованим списком також варто зберігати дані про вподобання користувачів для кожного об'єкта.

Завдяки інвертованому списку можна використати наступну схему вибірки даних:

- 1) для кожного вподобання користувача у сесії вибрати з інвертованого списку усі сесії, пов'язані з об'єктом вподобання;
- 2) вибрати вподобання користувачів усіх відібраних сесій.

Таким чином, це дозволяє переглядати лише частину даних. Завдяки цьому вибірка для формування рекомендації виконується швидше, але додавання нових вподобань потребує більше обчислень та пам'яті.

На рис. 3.14. показана схема реалізації розгорнутих зв'язаних списків у розробленій системі.

Для тестування запропонованого методу зберігання даних рекомендаційної системи використовувалася колаборативна фільтрація для формування списку рекомендацій. Колаборативна фільтрація – процес фільтрації інформації на основі схожості між користувачами або об'єктами

системи [35-38]. У її основі лежить кластеризація користувачів на основі різних методів, наприклад, коефіцієнтів подоби, що визначаються шляхом аналізу попередніх дій користувачів.

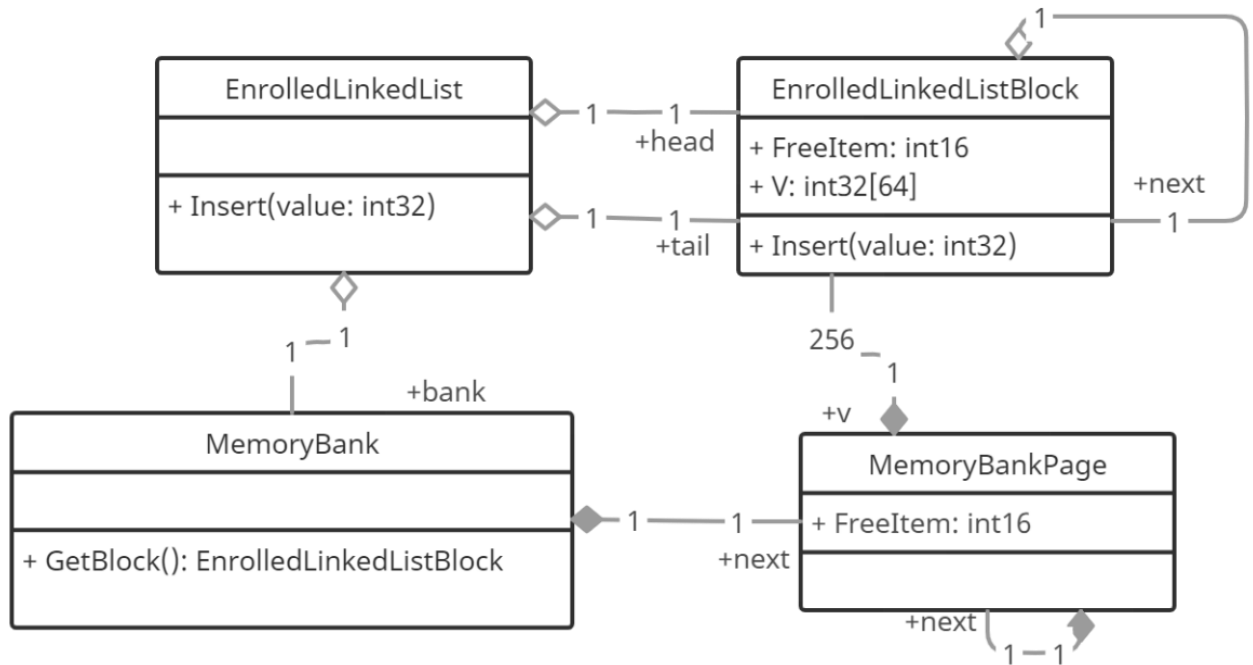


Рис. 3.14. Схема реалізації розгорнутих зв'язаних списків у розробленій системі

Було використано підхід User-based [38], за його застосування знаходяться близькі за уподобаннями користувачі, і вони «рекомендують» одному одному те, що вже спробував інший. Схожість користувачів визначалася на основі коефіцієнта кореляції Пірсона [39]. Даними для визначення схожості користувачів були лайки, які вони ставили об'єктам рекомендаційної системи.

На рис. 3.15 показана схема реалізації колаборативної фільтрації у розробленій системі.

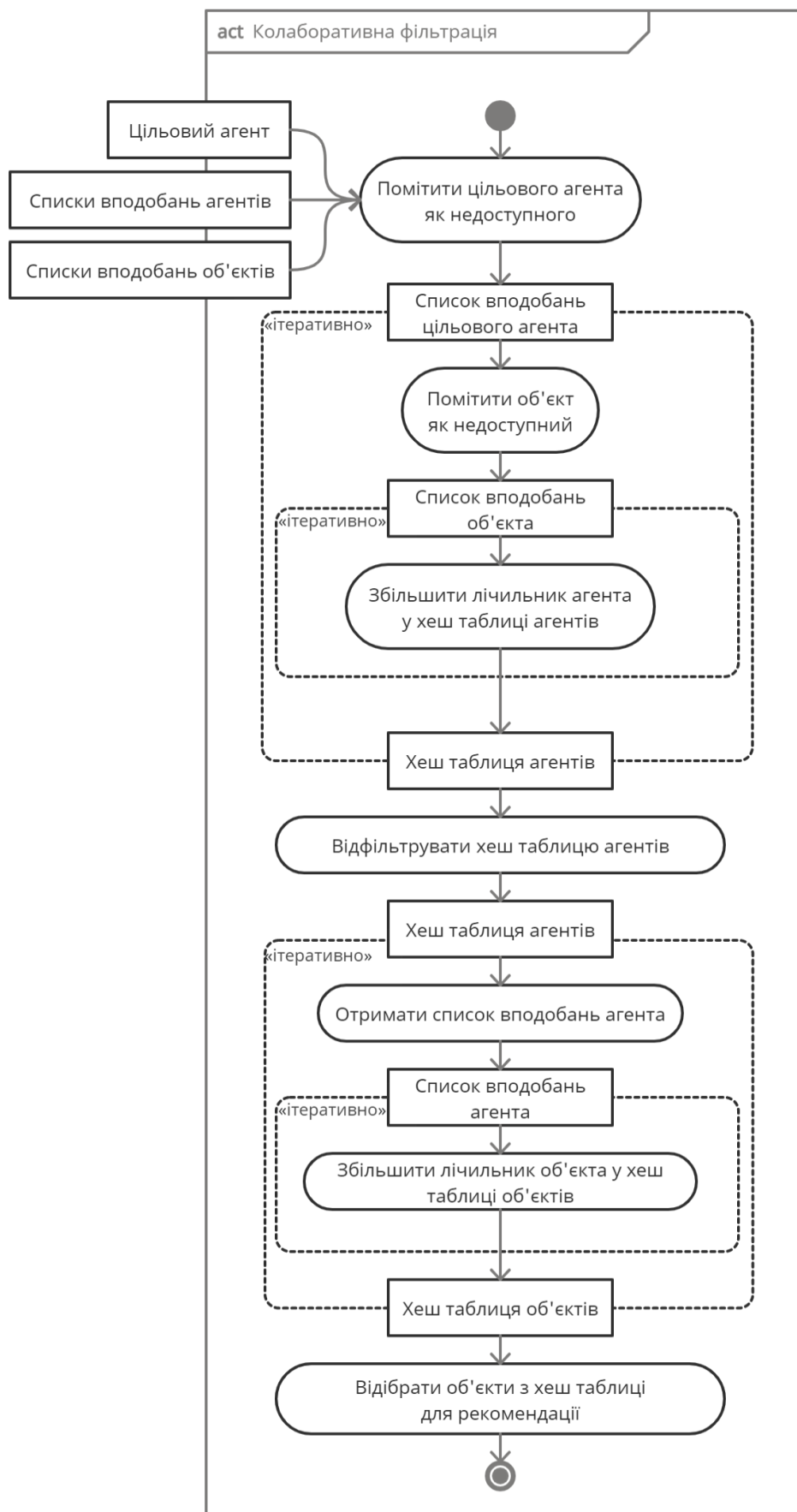


Рис. 3.15. Схема реалізації колаборативної фільтрації у розробленій системі

Етапи роботи запропонованого методу зберігання даних рекомендаційної системи в однорангових комп'ютерних мережах:

Етап 1. Інформація про користувачів та об'єкти рекомендаційної системи записуються у хеш-таблиці з відкритою адресацією. Ця хеш-таблиця також використовується для збереження проміжних значень, необхідних для генерації рекомендацій, у тому числі для підрахунку кількості входжень елементів. Для цього було реалізовано операцію інкременту, яка працює наступним чином:

- 1) здійснюється пошук елементу за ключем;
- 2) якщо елемент не існує, створюється новий елемент зі значенням 1;
- 3) якщо елемент існує і має значення -1, то його значення не змінюється, інакше – збільшується на 1.

Етап 2. Створюються розгорнуті зв'язані списки (зв'язні списки, що складаються із блоків з наперед визначеною кількістю елементів). Було використано цю структуру даних для збереження інформації про вподобання користувачів. Використовувалося два типи списків:

- 1) асоційовані з користувачем – містять список об'єктів, які він вподобав;
- 2) асоційовані з об'єктом – містять список користувачів, які вподобали цей об'єкт.

Дві копії інформації про вподобання дають можливість врахувати архітектуру P2P DHT мереж. Зокрема, кожен комп'ютер містить список вподобань, асоційований зі своїм користувачем, та списки вподобань, асоційовані з об'єктами, які на ньому розташовані. Ця інформація доступна для збору кожному комп'ютеру мережі при збереженні статистики дій відповідного користувача та завантажень наявних файлів. В такому разі під вподобанням буде розумітися факт завантаження файлу.

Етап 3. Доступ до потрібних даних, що не розміщені на комп'ютері користувача, якому треба сформувати рекомендації, може бути отриманий запитом до хеш-таблиць інших комп'ютерів P2P мережі за ідентифікаторами.

3.5.1. Проведення експериментів для порівняльного аналізу запропонованого методу зберігання даних рекомендаційної системи з існуючими

Було проведено експерименти для порівняльного аналізу запропонованого методу зберігання даних рекомендаційної системи з існуючими.

Для перевірки ефективності запропонованого методу зберігання даних було проведено експерименти з таким програмним забезпеченням:

- реляційна система управління базами даних Postgresql [40];
- резидентне сховище пар «ключ-значення» Redis [41];
- графова система управління базами Neo4j [26];
- розподілена система управління базами даних Cassandra [42];
- документо-орієнтована система управління базами даних MongoDB [43].

Кожен метод представлення даних було перевірено за наступними показниками:

- час заповнення сховища тестовими даними;
- об'єм пам'яті, зайнятий сховищем після заповнення;
- час генерації рекомендації.

Як тестові дані використано набір даних MovieLens [44]. Об'єм даних у форматі .csv складає 724 Мб. Результати проведених експериментів наведені у табл.3.7.

Час завантаження даних до програми, створеної на основі розробленого методу, склав 1,3 секунди, при цьому програма використала 616 Мб пам'яті. Генерація рекомендації для користувача зайняла 0,15 секунди.

У PostgreSQL було створено таблицю для представлення вподобань, яка складалася з полів для ідентифікаторів користувача та об'єкта. Було перевірено роботу без використання індексів та з використанням індексів для кожного поля. Додавання даних відбувалося за допомогою SQL-інструкцій,

при цьому було відзначено, що час додавання зменшувався при збільшенні кількості даних на один запит. Запис даних блоками по 100000 рядків склав 44 секунди у таблицю без індексів та 113 секунд у таблицю з індексами, тоді як запис блоками по 10000 – 58 та 125 секунд відповідно. Об'єм пам'яті, зайнятої таблицею, склав 786 Мб, індекси зайняли 331 Мб. Для побудови рекомендації було використано два варіанти SQL-запиту. У першому варіанті запиту було поєднано вирази IN та JOIN, час його виконання склав 13,9 секунд для таблиці без індексів та 16,3 секунд для таблиці з індексами. У другому варіанті запиту було використано лише JOIN, час виконання склав 14,5 секунд для таблиці без індексів та 6,7 секунд для таблиці з індексами.

У Redis для збереження даних було застосовано вбудований тип Set (множина). Для кожного користувача було створено множину з його вподобаннями, а для кожного об'єкта – множину з користувачами, які його вподобали. Redis здійснює запис даних асинхронно, і створення великої кількості одночасних запитів може вичерпати доступні ресурси. Тому великі об'єми даних потрібно завантажувати блоками, додаючи затримку після кожного блоку. Заповнення зайняло 89 секунд, для представлення даних Redis використав 1,4 Гб. Вибірка даних для генерації рекомендації зайняла 27 секунд.

Для завантаження даних до Neo4j було використано вбудовану функцію імпорту з .csv файлу. Завантаження вподобань зайняло 335 секунд, база даних зайняла 1,7 Гб. Для підвищення швидкодії було створено індекси по зв'язках між сутностями. Генерація рекомендацій зайняла 155 секунд.

Для роботи з Cassandra було використано імпорт з .csv файлу. Завантаження вподобань зайняло 6,83 секунд, база даних зайняла 1,01 Гб. Генерація рекомендацій зайняла 0,48 секунд.

В MongoDB дані було імпортовано також через .csv файл. Завантаження вподобань зайняло 3,08 секунд, база даних зайняла у пам'яті 1,31 Гб. Генерація рекомендацій зайняла 0,37 секунд.

Таблиця 3.7. Порівняння якості роботи запропонованої та існуючих систем управління базами даних

№	Інструмент	Час заповнення, с	Об'єм пам'яті, Гб	Час генерації рекомендації, с
1	PostgreSQL	113,00	1,09	6,70
2	Redis	89,00	1,40	27,00
3	Neo4j	335,00	1,70	155,00
4	Cassandra	6,83	1,01	0,48
5	MongoDB	3,08	1,31	0,37
6	Розроблена система на основі хеш-таблиць та лінійних списків	1,30	0,60	0,15

Як видно з табл. 3.7, запропонований метод зберігання даних рекомендаційної системи може зменшити витрати ресурсів часу та пам'яті. Зокрема, у порівнянні з найкращими результатами відомих методів, запропонований метод показав в 2,4 разів кращі результати по часу заповнення бази даних, в 1,7 разів – по використаному об'єму пам'яті та в 2,5 разів – по часу генерації рекомендацій.

Зрозуміло, що за це покращення доводиться платити розробкою власного продукту з урізаним функціоналом та власноруч організовувати підтримку та документацію.

Висновки до розділу 3

У роботі було проведено дослідження різних структур даних, які можна використати для зберігання даних рекомендаційної системи. Зокрема таких, як зв'язний список, розгорнутий зв'язний список, хеш-таблиця, В-дерево, В+-дерево та бінарні діаграми рішень.

Для проведення експериментів з порівняння ефективності застосування різних структур даних за затратами часу та пам'яті було розроблено програмну

модель рекомендаційної системи, в якій було виділено три основні сутності – агент, сесія та об’єкт. Найкращі результати показали методи зберігання даних із використанням розгорнутого та інвертованого розгорнутого зв’язних списків. Тому було вирішено також провести додаткову серію експериментів з цими структурами даних для різного розміру блоку списку. Розгорнутий список показав кращі результати за використовуваною пам’яттю в середньому в 1,54 разів та за часом ініціалізації сесій в середньому в 1,68 разів. Інвертований розгорнутий список показав перевагу за часом генерації рекомендацій в середньому в 1,43 разів. Час обробки лайків обидва методи показали приблизно однаковий. Профілювання показало, що 75% часу роботи тесту варіанта з розгорнутим списком зайняло генерування випадкових даних для програмного імітаційного моделювання агентів та об’єктів рекомендаційної системи, тож, саме сховище даних має високі показники ефективності. Профілювання варіанта із інвертованим списком показало, що доступ до випадкових блоків займає більше часу через неможливість закешувати їх, тож, за умов реального навантаження час вставки нових даних буде більшим, а відносна ефективність застосування інвертованого списку зросте. Для найбільш ефективного використання пам’яті розмір блоку зв’язного списку має бути адаптований таким чином, щоб блоки були максимально заповнені. Блоки малого розміру зменшують втрати пам’яті, але збільшують час обходу усіх елементів списку та збільшують накладні витрати пам’яті.

Також у цьому розділі було розроблено метод зберігання даних рекомендаційної системи на основі лінійних списків та хеш-таблиць, який можна використати для однорангових децентралізованих комп’ютерних мереж. Для перевірки ефективності запропонованого методу зберігання даних рекомендаційної системи було проведено порівняльні експерименти з існуючими методами. Кожен метод представлення даних було перевірено за наступними показниками: час заповнення сховища тестовими даними; об’єм пам’яті, зайнятий сховищем після заповнення; час генерації рекомендації.

Розроблена система на основі запропонованого методу значно випереджає розглянуті існуючі інструменти як за швидкістю роботи, так і за ефективністю використання пам'яті. Зокрема, у порівнянні з найкращими результатами відомих методів, запропонований метод показав в 2,4 разів кращі результати по часу заповнення бази даних, в 1,7 разів – по використаному об'єму пам'яті та в 2,5 разів – по часу генерації рекомендацій.

Список літератури до розділу 3

1. Міхав В. В., Мелешко Є. В., Якименко М. С., Шуліка Я. П. Розробка системи управління базою даних рекомендаційної системи для комп'ютерних та комп'ютерно-інтегрованих систем. *Центральноукраїнський науковий вісник. Технічні науки: зб. наук. пр.* Кропивницький: ЦНТУ, 2022. Вип. 5 (36). С. 130–136. DOI: 10.32515/2664-262X.2022.5(36).2.130-136 URL: [http://mapeia.kntu.kr.ua/pdf/5\(36\)_II/19.pdf](http://mapeia.kntu.kr.ua/pdf/5(36)_II/19.pdf) **(фахове видання категорії Б)**

2. Міхав В. В., Мелешко Є. В., Якименко М. С., Бащенко Д. В. Методи зберігання даних рекомендаційної системи на основі зв'язних списків. *Системи управління, навігації та зв'язку*. Полтава: ПНТУ, 2021. Т. 4(66). С. 59–62. DOI: 10.26906/SUNZ.2021.4.059.

URL: <http://journals.nupp.edu.ua/sunz/article/view/2422> **(фахове видання категорії Б)**

3. Міхав В. В., Мелешко Є. В., Якименко М. С. Метод зберігання даних рекомендаційної системи на основі бінарних діаграм рішень. *Системи управління, навігації та зв'язку*. Полтава: ПНТУ, 2020. Т. 2 (60). С. 85–89. DOI: 10.26906/SUNZ.2020.2.085. URL: <http://journals.nupp.edu.ua/sunz/article/view/1836> **(фахове видання категорії Б)**

4. Міхав В. В., Мелешко Є. В., Якименко М. С., Бащенко Д. В. Методи зберігання даних у рекомендаційних системах. *Цифрова трансформація суспільства: тези доп. І Міжнар. наук.-практ. конф. (Кропивницький, 21–22 квіт. 2022 р.)* / М-во освіти і науки України, Центральноукр. нац. техн. ун-

т. Кропивницький: ЦНТУ, 2022. С. 44–45.

5. Міхав В. В., Мелешко Є. В., Якименко М. С. Дослідження можливостей графової бази даних Neo4j для застосування у розробці рекомендаційних систем соціальних мереж. *Комбінаторні конфігурації та їхні застосування*: тези доп. XXIII Міжнар. наук.-практ. семінару ім. А. Я. Петренюка, присвяч. 70-річчю ЛА НАУ (Запоріжжя–Кропивницький, 13–15 трав. 2021 р.) / за ред. Г. П. Донця. Кропивницький: Ексклюзив-Систем, 2021. С. 124–127.

6. Міхав В. В., Мелешко Є. В. Застосування методу delta-encoding для представлення даних рекомендаційної системи. *Інформаційна безпека та комп'ютерні технології*: тези доп. IV Міжнар. наук.-практ. конф. (Кропивницький, 15–16 квіт. 2021 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2021. С. 57.

7. Міхав В. В., Мелешко Є. В. Порівняння стратегій редагування бінарних діаграм рішень для роботи з графовими даними. *Інформація, комунікація, суспільство*: тези доп. 9-ої Міжнар. наук. конф. (Львів, 21–23 трав. 2020 р.) / М-во освіти і науки України, Нац. ун-т «Львівська політехніка». Львів: Вид-во Львів. політехніки, 2020. С. 17–18.

8. Міхав В. В., Мелешко Є. В. Метод оптимізації швидкодії бінарних діаграм рішень при представленні даних рекомендаційної системи. *Інформаційна безпека та інформаційні технології*: тези доп. II Міжнар. наук.-практ. конф. (Кропивницький, 2–3 квіт. 2020 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2020. С. 17.

9. Міхав В. В. Алгоритми динамічного впорядкування змінних у впорядкованих бінарних діаграмах рішень. *Комбінаторні конфігурації та їх застосування*: тези доп. XVII Міжнар. наук.-практ. семінару (Кіровоград, 17–18 квіт. 2015 р.) / за ред. Г. П. Донця. Кропивницький: ЦНТУ, 2015. С. 75–78.

10. Міхав В. В. Програмне забезпечення для моделювання мереж репутації користувачів соціальних веб-ресурсів на основі бінарних діаграм

рішень. *Системи управління, навігації та зв'язку*. Полтава: ПНТУ, 2019. Т. 5 (57). С. 78–83. DOI: <https://doi.org/10.26906/SUNZ.2019.5.078>. URL: <http://journals.nupp.edu.ua/sunz/article/view/1720> **(фахове видання)**

11. Міхав В. В., Мелешко Є. В., Чабан О. О. Способи побудови рекомендаційних систем для соціальних мереж з врахуванням репутації користувачів. *Перспективні напрямки інформаційних і комп'ютерних систем та мереж, комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті*: тези доп. Міжнар. наук.-практ. конф. (Кропивницький, 13–14 лист. 2019 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2019. С. 86–87.

12. Drieieva H., Meleshko Ye., Drieiev O., Mikhav V. Computer simulation model of a computer network with fractal traffic for testing routing algorithms. *Advanced Information Systems*. 2022. No. 6 (4). P. 11–18. DOI: 10.20998/2522-9052.2022.4.02. URL: <http://ais.khpi.edu.ua/article/view/268306>. **(фахове видання категорії Б)**

13. Дреєва Г. М., Мелешко Є. В., Міхав В. В. Програмна імітаційна модель комп'ютерної мережі для тестування алгоритмів маршрутизації трафіку. *Автоматика, комп'ютерно-інтегровані технології та проблеми енергоефективності в промисловості і сільському господарстві*: матеріали Міжнар. наук.-техн. конф., (Кропивницький, 10–11 листоп. 2022 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: Ексклюзив-Систем, 2022. С. 44–45.

14. Meleshko Ye., Drieieva H., Drieiev O., Yakymenko M., Mikhav V., Shymko S. (2023). A Method of Routing of Fractal-like Traffic with Prediction of Router Load for Reduce the Probability of Network Packet Loss. *Computational Linguistics and Intelligent Systems (COLINS 2023)*: CEUR-WS, Kharkiv, Ukraine / O. Cherednichenko, L. Chyrun, V. Vysotska (Ed.). 2023. Vol. 3403(III). P. 434–448. (ISSN 1613-0073). DOI: 10.5281/zenodo.8231189. URL: <https://ceur-ws.org/Vol-3403/paper34.pdf>. **(SCOPUS)**

15. Drieieva H., Drieiev O., Meleshko Ye., Yakymenko M., Mikhav V. A method of determining the fractal dimension of network traffic by its probabilistic properties and experimental research of the quality of this method. *Computational Linguistics and Intelligent Systems (COLINS 2022)*: CEUR-WS, Gliwice, Poland / Vasyl Lytvyn (Ed.). 2022. Vol. 3154. P. 1694–1707. (ISSN 1613-0073). DOI: 10.5281/zenodo.7892006. URL: <https://ceur-ws.org/Vol-3171/paper120.pdf>. (SCOPUS)

16. Дреєва Г. М., Дреєв О. М., Мелешко Є. В., Якименко М. С., Міхав В. В. Дослідження якості роботи методу визначення фрактальної розмірності мережевого трафіку за його ймовірнісними властивостями. *Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління: тези доп. 12-ї Міжнар. наук.-техн. конф., (Баку–Харків–Жиліна, 27–28 квіт. 2022 р.)* / уклад. В. В. Косенко. Харків: ФОП Петров В. В., 2022. Т. 1. С. 126.

17. Shi L., Huang W. Apply social network analysis and data mining to dynamic task synthesis for persistent MMORPG virtual world. *International Conference on Entertainment Computing*. Berlin: Springer, 2004. P. 204–215.

18. Knuth D. Art of Computer Programming, The: Combinatorial Algorithms, Volume 4A, Part 1. *Addison-Wesley Professional; 1st edition*. 2011. 912 p.

19. Andersen H. R. An introduction to binary decision diagrams. *Lecture notes, available online, IT University of Copenhagen*. (1997) Vol. 5. URL: <https://tristable.net/coding/symbols/bdd97.pdf>

20. Drechsler R., Sieling D. Binary decision diagrams in theory and practice. *International Journal on Software Tools for Technology Transfer*. 2001. Vol. 3, P. 112–136. DOI: 10.1007/s100090100056

21. Brain M., Crawford S. The Importance of Memory Management in C. 2011. URL: <https://computer.howstuffworks.com/c-programming11.htm>.

22. Recommender Systems Handbook / F. Ricci, L. Rokach, B. Shapira, P. B. Kantor (eds.). New York, NY: Springer-Verlag New York, Inc., USA. 2010. 842 p.
23. Valois B.Jr.C., Oliveira M. A. Recommender systems in social networks. *JISTEM J.Inf.Syst. Technol. Manag.* 2011. Vol. 8 No. 3. P. 681–716. URL: https://www.scielo.br/scielo.php?script=sci_arttext&pid=S1807-17752011000300009
24. Fayyaz Z., Ebrahimian M., Nawara D., Ibrahim A., Kashef R. Recommendation systems: Algorithms, challenges, metrics, and business opportunities. *Advances in Artificial Intelligence: Machine Learning, Data Mining and Data Sciences*. 2020. Vol. 10(21), 7748. DOI: 10.3390/app10217748
25. Deldjoo Y., Schedl M., Cremonesi P., Pasi G. Recommender Systems Leveraging Multimedia Content. *ACM Computing Surveys*. 2020. Vol. 53, Issue 5, No. 106. pp. 1–38. URL: <https://dl.acm.org/doi/abs/10.1145/3407190>
26. Neo4j Documentation. *Official website of the graph database Neo4j*. 2021. URL: <https://neo4j.com/docs/>
27. Meleshko Ye. Computer model of virtual social network with recommendation system. *Scientific journal Innovative Technologies and Scientific Solutions for Industries*. Kharkiv: NURE, 2019. Iss. 2(8). P. 80–84.
28. Kumar A. Implementing real time recommendation systems using graph algorithms & exploring graph analytics in a graph database platform (neo4j). *Dissertation submitted in part fulfilment of the requirements for the degree of Master of Science in Data Analytics at Dublin Business School*. 2019. URL: <https://esource.dbs.ie/handle/10788/3708>
29. Kausar M. A., Nasar M. SQL versus NoSQL databases to assess their appropriateness for big data application. *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*. 2021. Vol. 14(4). P. 1098–1108.
30. Robinson I., Webber J., Eifrem E. Graph Databases. *O'Reilly Media*. 2013. 178 p.

31. Yi N., Li C., Feng X., Shi M. Design and implementation of movie recommender system based on graph database. *14th Web Information Systems and Applications Conference (WISA)*, IEEE. 2017. P. 132–135.
32. Сергієнко А. М., Молчанова А. А., Романкевич В. О. Комп'ютерна дискретна математика: навчальний посібник для здобувачів ступеня бакалавра за освітньою програмою «Інженерія програмного забезпечення комп'ютерних систем» спеціальності 121 «Інженерія програмного забезпечення» та освітньої програми: «Системне програмування та спеціалізовані комп'ютерні системи» спеціальності 123 «Комп'ютерні системи та мережі». Київ: НТУУ «КПІ ім. Ігоря Сікорського», 2022. 189 с.
33. Hunt J. J., Vo K. P., Tichy W. F. Delta algorithms: An empirical analysis. *ACM Transactions on Software Engineering and Methodology (TOSEM)*. 1998. Vol. 7(2). P. 192–214.
34. Selavo L. Dynamic data encoding for page-oriented memories. *Doctoral dissertation*. University of Pittsburgh. 2004.
35. Koren Y., Rendle S., Bell R. Advances in Collaborative Filtering. *In: Ricci, F., Rokach, L., Shapira, B. (eds). Recommender Systems Handbook*. New York, NY: Springer, 2022. DOI: 10.1007/978-1-0716-2197-4_3
36. Gupta M., Thakkar A., Aashish, Gupta V., Rathore D. P. S. Movie Recommender System Using Collaborative Filtering. *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India. 2020. P. 415–420. DOI: 10.1109/ICESC48915.2020.9155879.
37. Fan W., Ma Y., Yin D., Wang J., Tang J., Li Q. Deep social collaborative filtering. *In Proceedings of the 13th ACM Conference on Recommender Systems (RecSys '19)*. Association for Computing Machinery, New York, NY, USA, 2019. P. 305–313. DOI: 10.1145/3298689.3347011
38. Su X., Khoshgoftaar T. M. A survey of collaborative filtering techniques. *Advances in artificial intelligence*. 2009. URL: <https://downloads.hindawi.com/archive/2009/421425.pdf>

39. Zaiontz C. Basic Concepts of Correlation. *Real statistics using Excel*. URL: <https://real-statistics.com/correlation/basic-concepts-correlation/>
40. PostgreSQL Documentation. *Official website of the graph database PostgreSQL*. URL: <https://www.postgresql.org/docs/>
41. Redis Documentation. *Official website of the graph database Redis*. URL: <https://redis.io/docs/>
42. Cassandra Documentation. *Official website of the graph database Cassandra*. URL: <https://cassandra.apache.org/doc/latest/>
43. MongoDB Documentation. *Official website of the graph database MongoDB*. URL: <https://www.mongodb.com/docs/>
44. Harper F. M., Konstan J. A. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*. 2015. 19 p. URL: 10.1145/2827872

РОЗДІЛ 4

МЕТОД ПОШУКУ ТА ФІЛЬТРАЦІЇ ДАНИХ РЕКОМЕНДАЦІЙНОЮ СИСТЕМОЮ ДЛЯ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ КОРИСТУВАЧАМ ОДНОРАНГОВОЇ ДЕЦЕНТРАЛІЗОВАНОЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ

У цьому розділі розроблено аналітичну модель рекомендаційної системи для децентралізованої однорангової комп'ютерної мережі та спосіб формування рекомендацій користувачам з урахуванням кількості переходів до завантаження даних у P2P мережах [1].

Розроблено метод пошуку та фільтрації даних рекомендаційною системою для формування рекомендацій користувачам однорангової децентралізованої комп'ютерної мережі. Проведено експерименти для визначення якості роботи запропонованого методу пошуку та фільтрації даних рекомендаційною системою [2].

4.1. Розробка аналітичної моделі рекомендаційної системи для децентралізованої однорангової комп'ютерної мережі

Класичні моделі рекомендаційних систем розраховані на використання в ієрархічних комп'ютерних мережах [3, 4]. Тому для використання в однорангових комп'ютерних мережах їх треба адаптувати до архітектури цих мереж. Однорангові комп'ютерні мережі мають ряд особливостей, зокрема децентралізоване зберігання даних та постійні динамічні зміни у мережі [5-15], що потребує врахування при розробці програмного забезпечення для них.

Розглянемо однорангову комп'ютерну мережу, всі вузли якої знаходяться на зв'язку дротових мережевих систем чи в умовах впевненого прийому одне одного при використанні бездротового зв'язку. Для однозначної формалізації процесу моделювання приймемо, що мережа функціонує відповідно до алгоритмів багатостанційного доступу з контролем несучої і

запобіганням колізій. До кожного вузла мережі підключено декілька абонентів із унікальними адресами. Багато вузлів мережі та множина абонентів на кожному з вузлів можуть змінюватися. Мережеві налаштування обладнання виконані таким чином, що вузли мережі в кожний момент часу мають актуальну інформацію про інші вузли мережі та підключених до них абонентів. Це підвищить ефективність маршрутизації даних.

Для обміну рекомендаціями між вузлами мережі можна використовувати механізм розсилки інкрементальних повідомлень. Цей підхід передбачає періодичне (з періодом T) відправлення вузлами повідомлень, що містять інформацію про підключених до них абонентів. Отримавши таке повідомлення, вузли можуть визначити список своїх сусідів та список абонентів, підключених до цих сусідніх вузлів. Якщо вузол отримав повідомлення з іншого вузла, то вважає його своїм сусідом. Якщо вузол не отримував l поспіль повідомлень від іншого вузла протягом певного часу, він перестає вважати його своїм сусідом.

Повідомлення, що надсилаються вузлами, можуть бути одного з трьох типів:

- «Повний» – повідомлення, яке містить повний список абонентів вузла;
- «Зміна» – повідомлення, що містить інформацію про зміну складу абонентів вузла;
- «Порожній» – повідомлення, яке не містить інформації про склад абонентів.

Вибір типу повідомлення відбувається за наступними правилами. Повідомлення типу «Повний» генерується, якщо від моменту генерації попереднього «Повний» – повідомлення пройшло більше часу, ніж заданий поріг (згенеровано n повідомлень). Або якщо в мережі з'явився новий вузол, і до цього моменту було згенеровано менше f повідомлень «Повний», ніж заданий поріг. Повідомлення типу «Зміна» генерується, якщо після зміни складу абонентів вузла було згенеровано менше, ніж d повідомлень типу

«Зміна» або «Повний», і повідомлення типу «Зміна» містили інформацію тільки про цю зміну. В інших випадках генерується повідомлення типу «Порожній».

Припустимо, що вузли мережі мають доступ до актуальних даних про абонентів усіх сусідніх вузлів, які знаходяться в зоні їх дротового доступу або впевненого прийому. У такому разі ступенем актуальності будемо називати ймовірність P_{akt} того, що в будь-який момент часу всі вузли матимуть актуальні дані про абонентів усіх сусідніх вузлів. Важливо відзначити, що вузли можуть мати неактуальні дані про абонентів через наступні причини:

1. Змінився склад абонентів на деякому вузлі, але інші вузли не отримали відповідного повідомлення. І тут ймовірність того, що хоча б один вузол матиме актуальні дані, позначимо як $P_{akt}^{(1)}$.

2. З'явився новий вузол у мережі, і жоден з решти вузлів не знає, з якими абонентами пов'язаний цей вузол. При цьому новий вузол не знає абонентів інших вузлів. Імовірність того, що у випадковий момент часу хоча б один вузол не знає складу абонентів нового вузла та/або новий вузол не знає складу абонентів інших вузлів, позначимо як $P_{akt(n)}^{(1)}$.

3. Вузол покинув мережу, але решта вузлів все ще вважає його сусіднім. Імовірність того, що у випадковий момент часу вузли все ще вважають вузол, який залишив мережу, сусіднім, позначимо як $P_{akt(l)}^{(1)}$.

4. Деякий вузол мережі помилково виключено зі списку сусідніх. Імовірність того, що у випадковий момент часу хоча б один вузол мережі помилково не вважається сусіднім хоча б з одним вузлом, позначимо як $P_{akt(m)}^{(1)}$.

5. Оцінка ступеня актуальності P_{akt} робиться за допомогою аналітичного підходу, припускаючи, що всі причини зниження актуальності інформації, згадані вище, виникають незалежно одна від одної.

Таким чином, ступінь актуальності P_{akt} може бути оцінена наступним чином:

$$P_{akt} = (1 - P_{akt}^{(1)}) (1 - P_{akt(n)}^{(1)}) (1 - P_{akt(l)}^{(1)}) (1 - P_{akt(m)}^{(1)}). \quad (4.1)$$

Так як всі вузли мережі знаходяться в межах комунікаційної видимості один від одного та зони впевненого прийому, передача пакета може бути неуспішною лише через колізії. У середньому через час T_n змінюється склад вузлів мережі – по черзі відбувається одна з подій: в мережі з'являється новий вузол або один з вузлів покидає мережу. Також передбачається, що у середньому через T_h на кожному вузлі змінюється склад абонентів – по черзі відбувається одна з подій: підключається новий абонент або відключається один з підключених абонентів. В рамках цих припущень можна вважати, що кількість N вузлів та кількість N_h абонентів, підключених до кожного вузла, у середньому залишаються постійними.

Для визначення ймовірностей $P_{akt(l)}^{(1)}$, $P_{akt(n)}^{(1)}$, $P_{akt(l)}^{(1)}$ використовуються середні значення проміжків часу $t_{akt(l)}^{(1)}$, $t_{akt(n)}^{(1)}$, $t_{akt(l)}^{(1)}$ протягом яких інформація стає неактуальною після виникнення відповідних подій. Для обчислення ймовірності $P_{akt(m)}^{(1)}$ необхідно визначити, яку частку часу вузли мережі помилково не вважають деякий вузол сусіднім.

Введемо припущення, що середній час, протягом якого всі вузли дізнаються про склад абонентів нового вузла, не перевищує часу, протягом якого новий вузол дізнається про склад абонентів інших вузлів.

Далі було знайдено середній час неактуальності інформації $t_{akt(l)}^{(1)}$ при зміні складу абонентів біля деякого вузла та відсутності інформації про зміну. Для цього було введено наступну допоміжну величину, що синтезує ймовірність p успішної передачі інформаційних пакетів про зміни:

$$k_{i,j}(N, p) = \sum_{m=1}^j p_m(N, p), \quad (4.2)$$

де $p_m(N, p)$ – ймовірність того, що за m спроб передачі не всі N вузлів успішно передали свій пакет.

Відмітимо, що коефіцієнт $k_{i,j}$ знаходиться за допомогою виразу:

$$k_{i,j} = \sum_{r=1}^N \binom{N}{r} (-1)^{r+1} (1-p)^{ir} \frac{1-(1-p)^{jr-ir+\ell r}}{1-(1-p)^r}. \quad (4.3)$$

Величина $\mu_{\max}$ – математичне очікування випадкової величини, рівної максимуму з N незалежних випадкових величин μ_r , що мають рівномірний розподіл на відрізьку $[0, 1]$:

$$\mu_{\max} = M \left[\max_{r=1,2,\dots,N} \mu_r \right] = \frac{N}{N+1}. \quad (4.4)$$

Для розрахунку часу $t_{akt(l)}^{(1)}$, що потрібен для отримання даних про склад абонентів інших вузлів, розглянемо наступний сценарій. У розглянутому механізмі розсилки інкрементальних повідомлень вузли почнуть генерувати повідомлення «Повний» лише після того, як отримають перше повідомлення від нового вузла. Припустимо, що $t_n^{(1)}$ – середній час, протягом якого один вузол успішно передасть своє повідомлення. А $t_n^{(all)}$ – середній час, протягом якого всі вузли успішно передадуть свої повідомлення, що містять повний склад абонентів, починаючи з моменту успішної передачі новим вузлом свого повідомлення. Тоді розрахуємо час $t_{akt(l)}^{(1)}$ як суму:

$$t_{akt(l)}^{(1)} = t_n^{(1)} + t_n^{(all)}. \quad (4.5)$$

Новий вузол згенерує своє перше повідомлення в середньому через Δt після появи, наступне повідомлення буде згенероване через $2\Delta t$ після першого і так далі. Як було вказано при постановці завдання математичного моделювання, дуже важливим є врахування вимог достовірності та безпеки рекомендаційних повідомлень. Врахуємо ці характеристики шляхом введення в загальну модель показників:

– ймовірність достовірної передачі пакета $p_{saf}^{(M)}$, коли в мережі не відбуваються зміни (режим стабільної роботи);

– ймовірність безпечної передачі пакета $p_{sec}^{(M)}$, коли в мережі не відбуваються кібернебезпечних інцидентів (режим безпечної роботи). Враховуючи ці показники знайдемо час $t_n^{(1)}$, протягом якого новий вузол успішно передасть своє повідомлення, що складає:

$$t_n^{(1)} = 2\Delta t \left(\frac{1}{1 - (1 - p_{saf}^{(M)})(1 - p_{sec}^{(M)})} - \frac{1}{2} \right). \quad (4.6)$$

Після отримання першого повідомлення від нового вузла інші вузли згенерують m повідомлень «Повний». При цьому перше повідомлення «Повний» вузли згенерують протягом часу $2\Delta t \times \mu_{\max}$, друге повідомлення «Повний» – через $2\Delta t$ після першого, і так далі. Після того, як вузли згенерують m повідомлень «Повний», наступне повідомлення «Повний» буде згенеровано через $2\Delta t(n+1)$ після генерації останнього повідомлення «Повний» і так далі. Використовуючи коефіцієнт k_{ij} та ймовірність $p_m(N, p)$, що додані з початку припущень у виразі (4.2), отримуємо вираз для $t_n^{(all)}$.

$$t_n^{(all)} = 2\Delta t \left(\begin{aligned} &\mu_{\max}^{(N-1)} + k_{1,m-1} \left(N-1, \left(1 - (1 - p_{saf}^{(F)})(1 - p_{sec}^{(F)}) \right) \right) + \\ &+ \mu_{\max}^{(N-1)} (n-1) k_{m,m} \left(N-1, \left(1 - (1 - p_{saf}^{(F)})(1 - p_{sec}^{(F)}) \right) \right) + \\ &+ (n+1) k_{m+1,\infty} \left(n-1, \left(1 - (1 - p_{saf}^{(M)})(1 - p_{sec}^{(M)}) \right) \right) \end{aligned} \right),$$

де $1 - (1 - p_{saf}^{(F)})(1 - p_{sec}^{(F)})$ – ймовірність успішної передачі пакета, коли всі вузли мережі розсилають повідомлення «Повний».

Було проведено валідацію аналітичної моделі за допомогою моделювання в середовищі MathCad. Результати аналітичних розрахунків у вигляді графіків залежності $t_n^{(all)}$ від $p_{sec}^{(M)}$, $p_{saf}^{(M)}$ та Δt для різних значень N представлені на рис. 4.1, (а, б).

Експерименти показали, що запропонована аналітична модель, як мінімум, не суперечить основним положенням теорії інформації та передачі даних. Тому вона може бути використана для оцінки ступеня актуальності

мережевої інформації при різних параметрах рекомендаційної системи та механізмах розсилки інкрементальних повідомлень у ній.

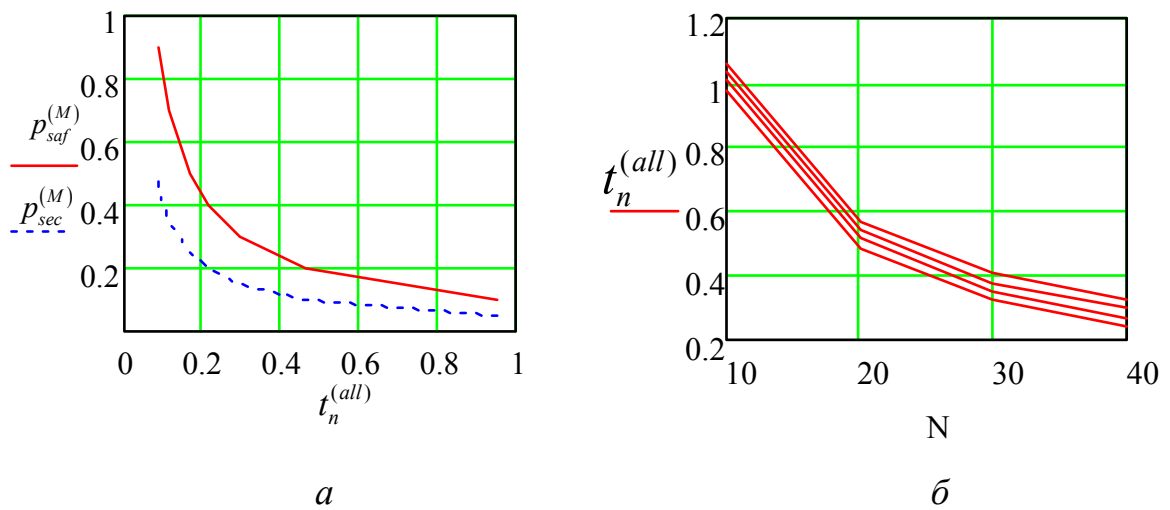


Рис. 4.1. Криві графіків залежності:

а) $t_n^{(all)}$ від значень $p_{sec}^{(M)}$, $p_{saf}^{(M)}$ при $\Delta t = 0.5$; б) $t_n^{(all)}$ від різних значень N

Запропонована модель може бути використана для вибору оптимальних параметрів механізму розсилки повідомлень, які максимізують актуальність рекомендаційних даних при заданих параметрах системи. Крім того, вона дозволяє проводити порівняльний аналіз різних механізмів розсилки рекомендаційних даних та встановлювати припустимі параметри рекомендаційної системи, при яких ступінь актуальності не падає нижче визначеного порогу.

4.2. Розробка способу формування рекомендацій для користувачів у P2P мережах з урахуванням кількості переходів до завантаження даних

Найпростіше побудувати рекомендаційну систему у централізованих однорангових мережах, оскільки там наявний сервер, через який проходить інформація про завантаження файлів. Отже, на основі статистики завантаження файлів різними користувачами можна побудувати

рекомендаційну систему на колаборативній фільтрації. Її точність можна підвищити, якщо надати можливість користувачам оцінювати файли та зберігати оцінки на сервері.

Більш складною задачею є побудова рекомендаційної мережі для децентралізованих P2P мереж – як неструктурованих так і структурованих. Оскільки сервери у таких системах відсутні, то кожен окремий комп'ютер буде самостійно розраховувати собі рекомендації на основі доступної йому інформації. Будемо вважати, що кожен комп'ютер системи містить інформацію про k найближчих (поняття та метрика визначення відстані залежить від конкретного алгоритму реалізації мережі) вузлів мережі, розташованих на них файлів та статистику завантажень (i , можливо, оцінки користувачів файлам). Також він може запитати інформацію у своїх «сусідів» про їх «сусідів», про яких йому не відомо і т. д. Чим більшою буде глибина запиту, тим більше буде інформації про наявні у мережі файли та тим точнішими будуть рекомендації, але й довшим – час формування рекомендацій та більшою – ймовірність виникнення проблеми «шторму трансляцій». Отже, при формуванні списку рекомендацій слід враховувати не тільки вподобання користувачів, а й кількість переходів до файлів у мережі.

Було запропоновано наступну модель роботи рекомендаційних систем у peer-to-peer мережах. У моделі присутні такі елементи:

$f_{i,k}$ – наявний файл k в бібліотеці користувача i ;

f_k – файл k в системі, незалежно від наявності його у користувача;

F_i – множина завантажених файлів користувача i ;

F – множина всіх фізично наявних файлів, файли дублюються у різних користувачів;

D – множина всіх оцінок користувачів файлам у системі (замість оцінки можна використовувати, наприклад, факт завантаження файлу, тоді D може приймати значення $[0, 1]$);

D_i – множина оцінок файлам користувача i ; $d_{i,k}$ – оцінка користувачем i окремого файлу k , $d_{i,k} \in [-1..1]$;

$v_{i,j}$ – коефіцієнт схожості вподобань користувачів i та j ;

p_i – ймовірність наявності в мережі користувача i в довільний момент часу;

$e(i, k)$ – наявність файлу k у користувача i (1 або 0);

$c_k(i, j)$ – мінімальна кількість переходів, яку потрібно здійснити від користувача i до користувача j ;

$c(i, k)$ – мінімальна кількість переходів від користувача i , до користувача, у якого наявний файл k ;

$U(k)$ – множина користувачів, які мають завантажений файл k ;

$|U|$ – потужність множини, кількість елементів в множині (приклад: $|U(k)|$ – кількість представлень файлу k);

N – кількість користувачів.

Нехай користувач i не має у власних списках F_i та D_i файлу k : $e(i, k)=0$, але відомо що цей файл є у наявності у користувача j . Тоді пропонуємо використати наступні величини для визначення необхідності рекомендувати цей файл поточному користувачу:

$c(i, j)>0$ – чим менше, тим краще, бо менший шлях до передавання інформації;

$v_{i,j}$ – чим більше, тим краще, бо більше шансів схожих вподобань у користувачів;

$d_{j,k}$ – чим більше, тим краще, бо файл k оцінено як хороший користувачем j ;

$|U(k)|$ – популярність файлу – чим більше, тим краще, бо розподілене отримання файлу значно скоротить час його завантаження. Впливає на оцінювання не безпосередньо, бо зі збільшенням цього числа буде більша множина рекомендацій від «сусідів».

Вага рекомендації від користувача j складатиме:

$$r(i, k, j) = \frac{v_{i,j} d_{j,k} p_j}{c(i, j)}. \quad (4.7)$$

Оцінкою швидкості завантаження файлу з одного джерела є:

$$t(i, k, j) \sim p_i p_j c(i, j), \quad (4.8)$$

де враховано ймовірність підключення до мережі двох користувачів одночасно та кількість сегментів, які потрібно подолати, щоб дістатися до файлу. Швидкість завантаження з групи джерел можна визначити наступним чином:

$$t(i, k) \sim \left(\sum_j^N \frac{e(j, k)}{p_i p_j c(i, j)} \right)^{-1}. \quad (4.9)$$

Алгоритм визначення рекомендації:

1) Обираємо глибину аналізування $n > 0$.

2) Для обраної глибини знаходимо множину рекомендаційних ваг для користувача i за виразом:

$$R_i = \bigcup_{k=1}^{|F|} R(i, k), \quad (4.10)$$

$$\text{де } R(i, k) = \sum_{j=1}^N r(i, k, j) \begin{cases} 1, & 1 \leq c(i, j) \leq n \\ 0, & c(i, j) > n \end{cases}.$$

3) Для користувача обираємо рекомендацію як $k_i = k \text{ of } \max(R_i)$. Якщо треба створити список рекомендацій – обираємо M найкращих і ранжуємо у порядку спадання значення R_i .

Підсилення або послаблення критерію рекомендації можна реалізувати коефіцієнтами $\alpha, \beta, \gamma, \delta$, якщо скористатися наступним виразом розрахунку ваги рекомендації:

$$r(i, k, j) = \frac{v_{i,j}^{\alpha} d_{j,k}^{\beta} p_j^{\gamma}}{c(i, j)^{\delta}}. \quad (4.11)$$

Розглянемо різні випадки, при яких у P2P мережах може виникнути необхідність надання користувачу списків рекомендацій:

1) користувач нічого не шукає, а тільки зайшов у додаток $\rightarrow$ надаємо користувачу список рекомендацій з врахуванням його вподобань.

2) користувач шукає конкретний файл, що був доданий до мережі раніше, за його ключем (хеш-значенням); файл не знайдено з різних причин (комп'ютери, що містять файл або таблиці маршрутизації до файлу, вийшли

з мережі, застосовуються технології P2P з ймовірнісним методом пошуку, наприклад, неструктуровані мережі та RBFS тощо) → надаємо користувачу список рекомендацій з врахуванням його вподобань та пошукового запиту.

3) користувач шукає файли по ключових словах (залежно від практичної реалізації файлообмінні системи можуть містити таку функцію, якщо зберігають не тільки файли, а й ключові слова); хеш-значення ключових слів, представлених у запиті, не знайдено у мережі з різних причин → надаємо користувачу список рекомендацій з врахуванням його вподобань та пошукового запиту. Або знайдено хеш-значення ключових слів, їм відповідає декілька різних файлів → формуємо пошукову видачу, ранжуємо список результатів рекомендаційною системою.

На основі запропонованої моделі пропонується наступний спосіб формування рекомендацій у P2P мережах з урахуванням кількості переходів до завантаження даних для (1) випадку використання:

1 Етап. Збирається інформація з таблиць маршрутизації комп'ютерів користувача та його «сусідів» і їх «сусідів» з глибиною пошуку n . Необхідною інформацією є хеш-значення файлів, їх розташування та статистика завантажень (та оцінки користувачів за наявності).

2 Етап. На основі зібраної інформації для кожного файлу, у створеному на кроці 1 списку, обчислюється R_i за формулою (4.11) з запропонованої математичної моделі, але тільки для файлів, які поточний користувач раніше не завантажував.

3 Етап. Обираємо M файлів з найвищим значенням R_i та ранжуємо у порядку спадання цього значення.

4 Етап. Надаємо користувачу список рекомендованих файлів, сформований на кроці 3.

5 Етап. Зберігаємо шлях до рекомендованих файлів у таблиці маршрутизації на час t_x для швидкого їх пошуку, на випадок, якщо протягом заданого проміжку часу користувач ними зацікавиться.

Запропонований спосіб формування рекомендацій для користувачів у P2P мережах з урахуванням кількості переходів до завантаження даних розрахований на випадок, коли користувачам слід надати загальні рекомендації незалежно від їх пошукового запиту. Рекомендації засновані на прогнозуванні вподобань користувачів, вподобання прогнозуються на основі зібраної інформації про дії схожих користувачів у системі. Також при формуванні вподобань враховується кількість переходів до потрібної інформації, щоб оптимізувати час формування рекомендацій та завантаження файлів.

4.3. Розробка методу пошуку та фільтрації даних рекомендаційною системою для формування рекомендацій користувачам однорангової децентралізованої комп'ютерної мережі

З урахуванням особливостей архітектури P2P DHT мереж було запропоновано метод пошуку та фільтрації даних рекомендаційною системою для формування рекомендацій користувачам однорангової децентралізованої комп'ютерної мережі.

Для реалізації рекомендаційної системи у P2P DHT мережі найкраще підійдуть підходи, засновані на колаборативній фільтрації, адже для неї достатньо знати тільки реакцію користувача на контент, що може бути представлена оцінками, лайками або самим фактом завантажень файлів. Вміст файлів, їх назви та інформацію про користувачів знати не обов'язково. Таким чином можна працювати лише з ідентифікаторами файлів, ідентифікаторами користувачів та статистикою завантаження файлів – це все можна отримати при роботі з децентралізованою структурованою одноранговою комп'ютерною мережею.

Для представлення даних рекомендаційної системи у пам'яті було використано дві структури даних: розгорнуті зв'язні списки та хеш-таблиці із

відкритою адресацією. Розгорнуті зв'язні списки були обрані після проведення експерименту з використанням програмної імітаційної моделі, де тестувалися різні методи зберігання даних рекомендаційної системи [16]. Хеш-таблиці обрані для проміжних обчислень.

Розгорнутий зв'язний список – це зв'язний список, який складається із блоків з наперед визначеною кількістю елементів. Така структура дає можливість поєднати ефективно розширення зв'язних списків та переваги послідовного доступу до елементів блоку. Було використано цю структуру даних для збереження інформації про вподобання користувачів.

Використовувалося два типи списків: 1) асоційовані з користувачем – містять список об'єктів, які він вподобав; 2) асоційовані з об'єктом – містять список користувачів, які вподобали цей об'єкт. Таким чином, використовуються дві копії інформації про вподобання, що збільшує витрати пам'яті, але дає можливість уникати перегляду усіх вподобань для генерації рекомендації та дає можливість врахувати архітектуру P2P DHT мереж. Зокрема, кожен комп'ютер може містити список вподобань асоційований зі своїм користувачем та списки вподобань асоційовані з об'єктами, які на ньому розташовані. Ця інформація доступна для збору кожному комп'ютеру, для цього треба тільки зберігати статистику дій відповідного користувача та статистику завантажень наявних файлів. В такому разі під вподобанням буде розумітися факт завантаження файлу. Якщо впровадити у P2P DHT мережу можливість оцінювання файлів, схема буде трохи складнішою. Доступ до потрібної інформації, яка не розміщена на комп'ютері користувача, якому треба сформулювати рекомендації, може бути отримано запитами до інших комп'ютерів за ідентифікаторами, організованими розподіленою хеш-таблицею P2P DHT мережі.

Також було використано хеш-таблиці з відкритою адресацією для збереження проміжних значень, необхідних для генерації рекомендацій, в

тому числі для підрахунку кількості входжень елементів. Для цього було реалізовано операцію інкременту, яка працює наступним чином:

1. Здійснюється пошук елементу за ключем.
2. Якщо елемент не існує, створюється новий елемент зі значенням, рівним 1.
3. Якщо елемент існує і має значення -1, то його значення не змінюється, інакше – збільшується на 1.

Такий підхід дає можливість заборонити інкрементувати певні елементи.

Запропонований метод пошуку та фільтрації даних рекомендаційною системою для формування рекомендацій користувачам однорангової децентралізованої комп'ютерної мережі складається з наступних етапів:

Етап 1. Створити хеш-таблицю з ідентифікаторами користувачів H_u та хеш-таблицю з ідентифікаторами об'єктів H_f . Одержати доступ до розгорнутих списків, асоційованих з користувачами L_u та асоційованих з об'єктами L_f , що містять історію дій (завантаження/вподобання).

Етап 2. Переглянути список вподобань L_{u0} користувача U_0 , для якого треба створити рекомендації. Для кожного вподобаного користувачем U_0 об'єкту створити у хеш-таблиці об'єктів H_f елемент зі значенням -1.

Етап 3. Для кожного користувача, який вподобав один із об'єктів системи, що в H_f мають значення -1, інкрементувати відповідний елемент у хеш-таблиці H_u .

Етап 4. Для кожного користувача U_i з хеш-таблиці користувачів H_u , елемент якого не містить -1, переглянути відповідний список вподобань L_{ui} та інкрементувати елементи у хеш-таблиці об'єктів H_f , якщо деякий користувач вподобав відповідний об'єкт.

Етап 5. Створити масив M_{rec} для збереження списку рекомендацій. Для кожного елементу з хеш-таблиці об'єктів H_f , значення якого не -1, здійснити

спробу вставити цей елемент до масиву M_{rec} у порядку спадання значень. Обрати $TopN$ рекомендацій з масиву M_{rec} для показу користувачу U_0 .

На початку роботи запропонований метод позначає цільового користувача U_0 (якому треба сформувати рекомендації) як недоступного для участі у проміжних обчисленнях, шляхом занесення у хеш-таблицю користувачів елемента, ключ якого – це ідентифікатор цього користувача, а значення дорівнює -1.

Після цього виконується обхід списку вподобань цільового користувача. Кожен вподобаний об'єкт заноситься до хеш-таблиці об'єктів зі значенням -1, щоб уникнути повторної рекомендації уже вподобаних об'єктів. Якщо система допускає повторні рекомендації, то цей етап може бути змінений.

Далі для кожного вподобаного об'єкту вибирається список користувачів, які його вподобали. Для кожного користувача із цього списку виконується інкрементація відповідного значення у хеш-таблиці користувачів. Це значення буде використано для визначення ступеня подібності користувачів. Після завершення обходу хеш-таблиця користувачів фільтрується, щоб виключити користувачів, вподобання яких не будуть використовуватися при формуванні рекомендації.

Під час наступного етапу виконується обхід усіх елементів хеш-таблиці користувачів. Для кожного з цих користувачів вибирається відповідний список вподобань. Для кожного елемента з цього списку виконується інкрементація відповідного значення у хеш-таблиці об'єктів. На основі цього значення визначається, чи потрапить об'єкт до рекомендації.

На останньому етапі з хеш-таблиці об'єктів відбираються елементи із найбільшими значеннями та передаються як рекомендації для цільового користувача.

4.4. Проведення експериментів для визначення якості роботи запропонованого методу пошуку та фільтрації даних рекомендаційною системою однорангової децентралізованої комп'ютерної мережі

Було проведено експериментальне дослідження запропонованого методу пошуку та фільтрації даних рекомендаційною системою для формування рекомендацій користувачам однорангової децентралізованої комп'ютерної мережі. Результати проведених експериментів наведено у табл. 4.1.

Для проведення експериментів використовувалася програмна імітаційна модель P2P DHT мережі та дані з датасету MovieLens [17], що переводилися до потрібного формату, зокрема, оцінки для простоти були перетворені на лайки в одній серії експериментів та лайки і дизлайки в іншій серії експериментів.

Були проведені наступні серії експериментів, що відрізнялися способом оцінювання об'єктів у програмній моделі та обробкою результатів:

- факт вподобання користувачем файлу визначався наявністю лайку (або ж інша інтерпретація – фактом завантаження файлу) – експерименти №1-3.

- ставлення користувача до файлу визначалося поставленими лайками та дизлайками – експерименти №4-6.

- ставлення користувача до файлу визначалося лайками та дизлайками, кожен із них множився на коефіцієнт $k=2$ перед запуском розрахунків для формування рекомендацій – експерименти №7-9.

- ставлення користувача до файлу визначалося лайками та дизлайками, кожен із них множився на коефіцієнт $k=5$ перед запуском розрахунків для формування рекомендацій – експерименти №10-12.

В межах кожної серії експериментів використовувалися різні розміри списків рекомендацій, а саме: 128, 512 та 1024 об'єкти у списку.

Для оцінки якості роботи рекомендаційної системи використовувалися наступні метрики [18]: Precision (точність), Recall (повнота) та RMSE (середньоквадратична помилка):

$$precision = \frac{tp}{tp + fp}, \quad (4.12)$$

$$recall = \frac{tp}{tp + fn}, \quad (4.13)$$

де tp – результати, в яких позитивний прогноз виявився вірним; fp – результати, в яких позитивний прогноз виявився помилковим; fn – результати, в яких негативний прогноз виявився помилковим.

$$RMSE = \sqrt{\frac{1}{|\tau|} \sum_{(u,i) \in \tau} (\hat{r}_{ui} - r_{ui})^2}, \quad (4.14)$$

де $\hat{r}_{ui}$ – прогнозовані рейтинги (вподобання) для тестового набору даних τ пар користувач-об'єкт (u, i) ; r_{ui} – справжні рейтинги.

Усі дані з датасету були поділені на робочі та тестові. На основі робочих даних здійснювалися прогнози вподобань користувачів та формувалися списки рекомендацій. На основі тестових даних перевірялася точність рекомендацій за формулами (4.12)-(4.14).

Таблиця 4.1. Результати експериментів для перевірки якості роботи запропонованого методу

№ експ.	Спосіб оцінювання контенту у системі	Кількість об'єктів у списку рекомендацій	Precision (Точність)	Recall (Повнота)	RMSE
1	Тільки лайки (або факт завантаження)	128	0,7605	0,2033	0,5258
2	Тільки лайки (або факт завантаження)	512	0,7160	0,4443	0,4471
3	Тільки лайки (або факт завантаження)	1024	0,6917	0,6074	0,4052

Продовження табл. 4.1

№ експ.	Спосіб оцінювання контенту у системі	Кількість об'єктів у списку рекомендацій	Precision (Точність)	Recall (Повнота)	RMSE
4	Лайки та дизлайки	128	0,7786	0,2082	0,5199
5	Лайки та дизлайки	512	0,7530	0,4346	0,4324
6	Лайки та дизлайки	1024	0,7382	0,5825	0,3812
7	Лайки та дизлайки, помножені на $k=2$	128	0,7936	0,1969	0,5219
8	Лайки та дизлайки, помножені на $k=2$	512	0,7820	0,3880	0,4399
9	Лайки та дизлайки, помножені на $k=2$	1024	0,7765	0,4983	0,3941
10	Лайки та дизлайки, помножені на $k=5$	128	0,8338	0,1507	0,5372
11	Лайки та дизлайки, помножені на $k=5$	512	0,8391	0,2293	0,4976
12	Лайки та дизлайки, помножені на $k=5$	1024	0,8401	0,2529	0,4858

Як показали результати експериментів, наведені у табл. 4.1, точність (Precision) запропонованого методу виявилася досить високою і варіювалася у діапазоні $0,6917 \div 0,8401$ залежно від обраних параметрів системи. Найкращі результати були отримані, коли була можливість використовувати і лайки, і дизлайки користувачів, а їх значення були помножені на коефіцієнт $k=5$.

Повнота (Recall) розробленого методу, на жаль, виявилася низькою через те, що метод часто робить помилкові негативні прогнози. Це жодним чином не впливає на точність рекомендацій, тому що об'єкти з негативним прогнозом вподобань не потрапляють до списку рекомендацій. Але це погіршує наповненість списків, адже багато елементів, які можна було додати до списків, відсіюються.

Середньоквадратична помилка досить низька, її значення варіювалося у діапазоні $0,3812 \div 0,5372$ і дещо зменшувалося при збільшенні довжини списку рекомендацій.

Висновки до розділу 4

У цьому розділі було розроблено аналітичну модель рекомендаційної системи для децентралізованої однорангової комп'ютерної мережі та спосіб формування рекомендацій користувачам із урахуванням кількості переходів до завантаження даних у P2P мережах.

Розроблено метод пошуку та фільтрації даних рекомендаційною системою для формування рекомендацій користувачам однорангової децентралізованої комп'ютерної мережі.

Також було проведено експерименти для визначення якості роботи запропонованого методу пошуку та фільтрації даних рекомендаційною системою. Результати експериментів показали, що розроблений метод має досить високі показники точності, а саме, за метрикою Precision, – від 0,6917 до 0,8401, залежно від обраних параметрів системи, та низьку середньоквадратичну помилку – від 0,3812 до 0,5372. Найкращі результати отримані, коли була можливість використовувати і лайки, і дизлайки користувачів, а їх значення множилися на коефіцієнт $k=5$. Недоліком розробленого методу є низькі показники повноти за метрикою Recall, що виникає через те, що метод часто робить помилкові негативні прогнози. Це не впливає на точність рекомендацій, але погіршує наповненість списків рекомендацій, адже багато елементів, які можна було додати до них, відсіюються. Повнотою рекомендацій було пожертвовано для адаптації колаборативної фільтрації до архітектури P2P мереж.

Список літератури до розділу 4

1. Mikhav V., Semenov S., Meleshko Y., Yakymenko M., Shulika Y. Constructng the mathematical model of a recommender system for decentralized peer-to-peer computer networks. *Eastern-European Journal of Enterprise Technologies*. 2023. Vol. 4 (9(124)). P. 24–35. (ISSN 1729-3774). DOI: 10.15587/1729-4061.2023.286187. URL: <https://journals.uran.ua/eejet/article/view/286187> **(SCOPUS)**
2. Міхав В. В., Мелешко Є. В. Метод роботи рекомендаційної системи у комп'ютерній мережі типу peer to peer. *Системи управління, навігації та зв'язку*. Полтава: ПНТУ, 2023. Т. 1(71). С. 112–117. DOI: 10.26906/SUNZ.2023.1.112. URL: <http://journals.nupp.edu.ua/sunz/article/view/2839> **(фахове видання категорії Б)**
3. Recommender Systems Handbook / Ed. F. Ricci, L. Rokach, B. Shapira, P. B. Kantor. New York, NY, USA: Springer-Verlag New York, Inc. 2010. 842 p.
4. Jones M. Recommender systems, Part 1. In-troduction to approaches and algorithms. Learn about the concepts that underlie web recommendation engines. 2013. URL: https://www.ibm.com/developerworks/opensource/library/os-recommender1/index.html?s_tact=105agx99&s_cmp=cp
5. Riposo Ju. Diffusion on the Peer-to-Peer Network. *LAP LAMBERT Academic Publishing*. 2022. 100 p.
6. Koo S. G. M. Multimedia Content Distribution Using Peer-to-Peer Overlay Networks: The Design and Analysis of the Next Generation Peer-to-Peer Networks. VDM Verlag Dr. Müller. 2008. 88 p.
7. Milojicic D. S., Kalogeraki V., Lukose R., Nagaraja K., Pruyne J., Richard B., Rollins S., Xu Z. Peer-to-peer computing. *Technical Report HPL-2002-57, HP Labs*. 2002. 51 p. URL: <https://www.cs.kau.se/cs/education/courses/dvad02/p2/seminar4/Papers/HPL-2002-57R1.pdf>

8. Zeinalipour-Yazti D., Kalogeraki V., Gunopulos D. Information retrieval techniques for peer-to-peer networks. *Computing in Science & Engineering*, 2004. Vol. 6. No. 4. P. 20–26. DOI: 10.1109/MCSE.2004.12
9. Lua E. K., Crowcroft J., Pias M., Sharma R., Lim S. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. *IEEE Communications survey and tutorial*. 2004. URL: <https://snap.stanford.edu/class/cs224w-readings/lu04p2p.pdf>
10. Kalogeraki V., Gunopulos D., Zeinalipour-Yazti D. A Local Search Mechanism for Peer-to-Peer Networks. *Proc. of CIKM'02*, McLean VA, USA, 2002. URL: <http://alumni.cs.ucr.edu/~csyiazti/downloads/papers/cikm02/cikm02.pdf>
11. Zeinalipour-Yazti D. Information Retrieval in Peer-to-Peer Systems. *M.Sc Thesis, Dept. of Computer Science, University of California Riverside*. 2003. URL: <http://alumni.cs.ucr.edu/~csyiazti/papers/msc/html/>
12. Kademlia: A Design Specification. 2010. URL: <https://xlattice.sourceforge.net/components/protocol/kademlia/specs.html>
13. Stoica I., Morris R., Karger D. R., Kaashoek M. F., Balakrishnan H. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. *ACM SIGCOMM Computer Communication Review*. 2001. Vol. 31(4). DOI: 10.1145/964723.383071
14. The BitTorrent Protocol Specification. 2017. URL: http://www.bittorrent.org/beps/bep_0003.html
15. Gnutella Protocol Development. 2003. URL: <https://rfc-gnutella.sourceforge.net>
16. Міхав В. В., Мелешко Є. В., Якименко М. С., Бащенко Д. В. Методи зберігання даних рекомендаційної системи на основі зв'язних списків. *Системи управління, навігації та зв'язку*. Полтава: ПНТУ, 2021. Т. 4(66). С. 59–62. DOI: 10.26906/SUNZ.2021.4.059
17. Harper F. M., Konstan J. A. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*. 2015. 19 p. URL: 10.1145/2827872

18. Мелешко Є. В. Методи оцінки якості роботи рекомендаційних систем. *Системи управління, навігації та зв'язку*. Полтава: ПНТУ, 2018. Вип. 5 (51). С. 92–97. DOI: 10.26906/SUNZ.2018.5.092

ОСНОВНІ ВИСНОВКИ

У дисертаційній роботі вирішена актуальна науково-практична задача, що полягає у підвищенні ефективності методів обробки даних в однорангових децентралізованих комп'ютерних мережах шляхом розробки й удосконалення моделей і методів роботи рекомендаційних систем для цієї архітектури мереж.

Проведено дослідження та порівняльний аналіз методів роботи peer-to-peer комп'ютерних мереж, методів зберігання та пошуку даних у них, а також моделей децентралізованих рекомендаційних мереж.

Проведене дослідження показало, що P2P мережі знову актуальні завдяки бурхливому розвитку блокчейну, Інтернет-телебаченню і стрімінгу, децентралізованому зберігання даних тощо, а рекомендаційні системи дозволяють полегшити пошук при великій кількості неструктурованого контенту, доповнюючи або заміняючи класичну пошукову видачу рекомендаціями. В P2P мережах застосування рекомендаційних систем також має додаткову користь. Через проблеми індексації і пошуку раніше додані файли можуть бути недоступні для користувача P2P мережі. У такому випадку замість відсутнього файлу користувачу можна надати список рекомендацій з врахуванням його вподобань і пошукового запиту. Також при здійсненні пошуку і створенні рекомендацій користувачам важливо враховувати достовірність та інформаційну безпеку і час на доступ до інформації, а також знаходити баланс між точністю і швидкістю отримання результатів. Важливим питанням є також кількість пам'яті, яку можна виділити під дані рекомендаційної системи, адже у децентралізованій P2P мережі їх доведеться зберігати розподілено на комп'ютерах учасників.

Основні наукові та практичні результати дисертаційної роботи:

1. Вперше розроблено математичну модель процесів збору і обробки даних для рекомендаційної системи в одноранговій децентралізованій комп'ютерній мережі, яка відрізняється від відомих можливістю оцінки

ймовірісно-часових характеристик процесів формування і зміни рекомендацій за допомогою мультитригерного GERT-моделювання та врахуванням вимог достовірності і безпеки даних під час змін у структурі мережі, що дозволяє здійснювати раціональний вибір параметрів системи. Отримано аналітичний вираз, за допомогою якого є можливість оцінити щільність розподілу ймовірностей часу ідентифікації стану вузлів рекомендаційної системи. Використання мультитригерного підходу для GERT-моделювання дозволило, в порівнянні з відомими моделями, підвищити точність результатів моделювання до 5%. Також розроблено алгоритми програмного імітаційного моделювання рекомендаційної системи однорангової децентралізованої комп'ютерної мережі та процесів у ній, що дають можливість проводити тестування різних методів зберігання, пошуку та фільтрації даних рекомендаційної системи у P2P мережі.

2. Удосконалено метод зберігання даних рекомендаційної системи, який відрізняється від відомих адаптацією до архітектури однорангових децентралізованих комп'ютерних мереж, використанням хеш-таблиць для зберігання даних про користувачів і об'єкти контенту та зв'язних списків для зберігання даних про рекомендації, що дало змогу зменшити витрати часу і пам'яті на процеси обробки даних системи. Розроблено відповідні алгоритми зберігання даних рекомендаційної системи однорангової децентралізованої комп'ютерної мережі, що дозволяють зменшити витрати пам'яті і часу на процеси зберігання та читання даних системи. Зокрема, розроблена система на основі запропонованих алгоритмів при використанні у децентралізованій одноранговій мережі показує наступні результати в порівнянні з найкращими результатами відомих систем: у 2,4 разів кращі результати по часу заповнення бази даних, в 1,7 разів кращі результати по використаному об'єму пам'яті, в 2,5 разів кращі результати по часу генерації рекомендацій.

3. Удосконалено метод пошуку та фільтрації даних рекомендаційною системою для формування рекомендацій користувачам, який відрізняється від відомих використанням запропонованого методу зберігання даних для

збереження проміжних та підсумкових результатів обчислень, що дозволило використовувати його в однорангових децентралізованих комп'ютерних мережах та зменшити витрати часу і пам'яті при забезпеченні високої точності прогнозування вподобань користувачів. Розроблено відповідні алгоритми пошуку та фільтрації даних рекомендаційною системою для формування рекомендацій користувачам в однорангових децентралізованих комп'ютерних мережах, які зменшують витрати часу і пам'яті при забезпеченні високої точності прогнозування вподобань користувачів. Точність розробленого алгоритму створення рекомендацій сягає до 0,84, в залежності від обраних параметрів системи. Проведено оцінку якості та ефективності запропонованих методів і моделей шляхом проведення експериментів на програмній імітаційній моделі.

Практичне значення отриманих результатів підтверджено відповідними актами впровадження. Результати дисертаційних досліджень впроваджені і використовуються у діяльності ІТ-компанії ТОВ "ОНІКС-СИСТЕМЗ", а також їх використано у навчальному процесі Центральноукраїнського національного технічного університету.

Таким чином, сукупність отриманих у дисертаційній роботі наукових результатів і одержана в ході проведення експериментальних досліджень оцінка їх ефективності дозволяють вважати сформульовану науково-практичну задачу підвищення ефективності методів обробки даних в однорангових децентралізованих комп'ютерних мережах шляхом розробки і удосконалення моделей і методів роботи рекомендаційних систем для цієї архітектури мереж – вирішеною, а поставлену мету зі зменшення витрат часу та пам'яті на роботу рекомендаційної системи в однорангових децентралізованих комп'ютерних мережах при достатній точності створення рекомендацій користувачам – досягнутою.

ДОДАТКИ

Додаток А. Акти впровадження дисертаційних досліджень

ЗАТВЕРДЖУЮ:

Проректор з наукової роботи

Центральноукраїнського національного
технічного університету

Олександр ЛЕВЧЕНКО

с. 23 2022р.



АКТ

про впровадження результатів дисертаційної роботи

Міхава Володимира Володимировича

"МОДЕЛЬ ТА МЕТОДИ РОБОТИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ У
КОМП'ЮТЕРНИХ МЕРЕЖАХ ТИПУ PEER-TO-PEER"

на здобуття наукового ступеня доктора філософії
за спеціальністю 123 «Комп'ютерна інженерія»

Комісія у складі: голови – завідуючого кафедрою кібербезпеки та програмного забезпечення Центральноукраїнського національного технічного університету доктора технічних наук, професора Смірнова О.А. та членів комісії – доцента кафедри кібербезпеки та програмного забезпечення, кандидата технічних наук, доцента Минайленка Р.М. та доцента кафедри кібербезпеки та програмного забезпечення, кандидата технічних наук, доцента Боська В.В., склала цей акт про те, що при розробці лекційних, практичних та лабораторних занять з навчальних дисциплін «Проектування комп'ютерних систем та мереж» та «Алгоритми та структури даних» у навчальному процесі Центральноукраїнського національного технічного університету були використані наступні результати наукових досліджень Міхава Володимира Володимировича:

– Удосконалено метод зберігання даних рекомендаційної системи, який відрізняється від відомих адаптацією до архітектури однорангових децентралізованих комп'ютерних мереж та використанням хеш-таблиць для зберігання даних про користувачів і об'єкти контенту та зв'язних списків для зберігання даних про рекомендації, що дозволило зменшити витрати часу і пам'яті на процеси обробки даних системи.

– Удосконалено метод пошуку та фільтрації даних

рекомендаційною системою для формування рекомендацій користувачам, який відрізняється від відомих використанням запропонованого методу зберігання даних для збереження проміжних та підсумкових результатів обчислень, що дозволило використовувати його в однорангових децентралізованих комп'ютерних мережах та мінімізувати витрати часу і пам'яті при забезпеченні високої точності прогнозування вподобань користувачів.

Застосування результатів дисертаційних досліджень Міхава Володимира Володимировича дозволило підвищити рівень засвоєння навчального матеріалу з дисциплін «Проектування комп'ютерних систем та мереж» та «Алгоритми та структури даних» за рахунок більш поглибленого вивчення сучасних та перспективних методів роботи рекомендаційних систем.

Голова комісії

завідувач кафедри кібербезпеки
та програмного забезпечення
Центральноукраїнського національного
технічного університету
доктор технічних наук, професор



Олексій СМІРНОВ

Члени комісії:

доцент кафедри кібербезпеки
та програмного забезпечення
кандидат технічних наук, доцент



Роман МИНАЙЛЕНКО

доцент кафедри кібербезпеки
та програмного забезпечення
кандидат технічних наук, доцент



Віктор БОСЬКО

ЗАТВЕРДЖУЮ:

Директор ТОВ «ОНІКС-СИСТЕМЗ»

Роман СЕРЕБРЯКОВ


(підпис)

«22» грудень 2022 року

АКТ

про впровадження результатів дисертаційної роботи
аспіранта кафедри кібербезпеки та програмного забезпечення Центрально-
українського національного технічного університету
Міхава Володимира Володимировича
на тему "МОДЕЛЬ ТА МЕТОДИ РОБОТИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ
У КОМП'ЮТЕРНИХ МЕРЕЖАХ ТИПУ PEER-TO-PEER"
на здобуття наукового ступеня доктора філософії
за спеціальністю 123 «Комп'ютерна інженерія»

Комісія у складі голови – директора ТОВ «ОНІКС-СИСТЕМЗ» Серебрякова Р.М. та членів комісії – провідного розробника ТОВ «ОНІКС-СИСТЕМЗ» Гордієнко В.В. і провідного розробника ТОВ «ОНІКС-СИСТЕМЗ» Дабич А.В., склала цей акт про те, що у діяльності ТОВ «ОНІКС-СИСТЕМЗ» реалізовано результати наукових досліджень Міхава Володимира Володимировича.

Впроваджено у виробництво удосконалений метод зберігання даних рекомендаційної системи, який відрізняється від відомих адаптацією до архітектури однорангових децентралізованих комп'ютерних мереж та використанням хеш-таблиць для зберігання даних про користувачів і об'єкти контенту і зв'язних списків для зберігання даних про рекомендації. Отриманий корисний ефект від впровадження запропонованого методу зберігання даних

рекомендаційної системи полягає у зменшенні витрат часу і пам'яті на процеси обробки даних системи.

Голова комісії

Директор ТОВ

«ОНІКС-СИСТЕМЗ»



Роман СЕРЕБРЯКОВ

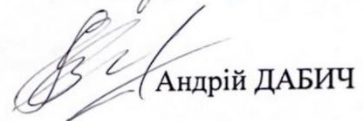
Члени комісії:

провідний розробник



Володимир ГОРДІЄНКО

провідний розробник



Андрій ДАБИЧ

**Додаток Б. Список публікацій здобувача за темою дисертації
та відомості про апробацію результатів дисертації**

Список публікацій здобувача за темою дисертації

1. Mikhav V., Semenov S., Meleshko Y., Yakymenko M., Shulika Y. Constructng the mathematical model of a recommender system for decentralized peer-to-peer computer networks. *Eastern-European Journal of Enterprise Technologies*. 2023. Vol. 4 (9(124)). P. 24–35. (ISSN 1729-3774). DOI: 10.15587/1729-4061.2023.286187. URL: <https://journals.uran.ua/eejet/article/view/286187> (SCOPUS)
2. Meleshko Ye., Drieieva H., Drieiev O., Yakymenko M., Mikhav V., Shymko S. (2023). A Method of Routing of Fractal-like Traffic with Prediction of Router Load for Reduce the Probability of Network Packet Loss. *Computational Linguistics and Intelligent Systems (COLINS 2023)*: CEUR-WS, Kharkiv, Ukraine / O. Cherednichenko, L. Chyrun, V. Vysotska (Ed.). 2023. Vol. 3403(III). P. 434–448. (ISSN 1613-0073). DOI: 10.5281/zenodo.8231189. URL: <https://ceur-ws.org/Vol-3403/paper34.pdf>. (SCOPUS)
3. Drieieva H., Drieiev O., Meleshko Ye., Yakymenko M., Mikhav V. A method of determining the fractal dimension of network traffic by its probabilistic properties and experimental research of the quality of this method. *Computational Linguistics and Intelligent Systems (COLINS 2022)*: CEUR-WS, Gliwice, Poland / V. Lytvyn (Ed.). 2022. Vol. 3154. P. 1694–1707. (ISSN 1613-0073). DOI: 10.5281/zenodo.7892006. URL: <https://ceur-ws.org/Vol-3171/pa-per120.pdf>. (SCOPUS)
4. Мелешко Є. В., Дреєв О. М., Лавданський А. О. Модель рекомендаційної системи для комп'ютерних мереж типу P2P. *Вісник Черкаського державного технологічного університету*. 2023. Вип. 1. С. 52–60.

DOI: 10.24025/2306-4412.1.2023.273495. URL: <http://vtn.chdtu.edu.ua/article/view/273495> **(фахове видання категорії Б)**

5. Міхав В. В., Мелешко Є. В. Метод роботи рекомендаційної системи у комп'ютерній мережі типу peer to peer. *Системи управління, навігації та зв'язку*: ПНТУ, Полтава. 2023. Т. 1(71). С. 112–117. DOI: 10.26906/SUNZ.2023.1.112. URL: <http://journals.nupp.edu.ua/sunz/article/view/2839> **(фахове видання категорії Б)**

6. Drieieva H., Meleshko Ye., Drieiev O., Mikhav V. Computer simulation model of a computer network with fractal traffic for testing routing algorithms. *Advanced Information Systems*. 2022. No. 6 (4). P. 11–18. DOI: 10.20998/2522-9052.2022.4.02. URL: <http://ais.khpi.edu.ua/article/view/268306>. **(фахове видання категорії Б)**

7. Міхав В. В., Мелешко Є. В., Якименко М. С., Шуліка Я. П. Розробка системи управління базою даних рекомендаційної системи

8. для комп'ютерних та комп'ютерно-інтегрованих систем. *Центральноукраїнський науковий вісник. Технічні науки: зб. наук. пр.*: ЦНТУ, Кропивницький, 2022. Вип. 5 (36). С. 130–136. DOI: 10.32515/2664-262X.2022.5(36).2.130-136 URL: [http://mapiea.kntu.kr.ua/pdf/5\(36\)_II/19.pdf](http://mapiea.kntu.kr.ua/pdf/5(36)_II/19.pdf) **(фахове видання категорії Б)**

9. Міхав В. В., Мелешко Є. В., Якименко М. С., Бащенко Д. В. Методи зберігання даних рекомендаційної системи на основі зв'язних списків. *Системи управління, навігації та зв'язку*: ПНТУ, Полтава. 2021. Т. 4(66). С. 59– 62. DOI: 10.26906/SUNZ.2021.4.059. URL: <http://journals.nupp.edu.ua/sunz/article/view/2422> **(фахове видання категорії Б)**

10. Міхав В. В., Мелешко Є. В., Якименко М. С. Метод зберігання даних рекомендаційної системи на основі бінарних діаграм рішень. *Системи управління, навігації та зв'язку*: ПНТУ, Полтава. 2020. Т. 2 (60). С. 85–89.

DOI: 10.26906/SUNZ.2020.2.085. URL: <http://journals.nupp.edu.ua/sunz/article/view/1836> **(фахове видання категорії Б)**

11. Міхав В. В. Програмне забезпечення для моделювання мереж репутації користувачів соціальних веб-ресурсів на основі бінарних діаграм рішень. *Системи управління, навігації та зв'язку*: ПНТУ, Полтава, 2019. Т. 5 (57). С. 78–83. DOI: <https://doi.org/10.26906/SUNZ.2019.5.078>. URL: <http://journals.nupp.edu.ua/sunz/article/view/1720> (**фахове видання**)

Відомості про апробацію результатів дисертації

1. Міхав В. В., Мелешко Є. В., Лавданський А. О. Дослідження принципів роботи однорангових децентралізованих структурованих комп'ютерних мереж. *Інформаційна безпека та комп'ютерні технології*: тези доп. VI Міжнар. наук.-практ. конф. (Кропивницький, 20–21 квіт. 2023 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2023. С. 51–52.

2. Міхав В. В., Мелешко Є. В., Дреєв О. М., Миронець І. В. Модель рекомендаційної системи для комп'ютерної мережі типу peer to peer. *Інновації та перспективні шляхи розвитку інформаційних технологій (ІПШРІТ-2022)*: тези доп. Міжнар. наук.-практ. Інтернет-конф., (Черкаси, 9 груд. 2022 р.) / упоряд.: Т. О. Прокопенко, Я. В. Тарасенко. М-во освіти і науки України, Черкас. держ. технол. ун-т, 2022. – С. 59–60.

3. Дреєва Г. М., Дреєв О. М., Мелешко Є. В., Якименко М. С., Міхав В. В. Дослідження якості роботи методу визначення фрактальної розмірності мережевого трафіку за його ймовірнісними властивостями. *Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління*: тези доп. 12-ї Міжнар. наук.-техн. конф., (Баку–Харків–Жиліна, 27–28 квіт. 2022 р.) / уклад. В. В. Косенко. Харків: ФОП Петров В. В., 2022. Т. 1. С. 126.

4. Дреєва Г. М., Мелешко Є. В., Міхав В. В. Програмна імітаційна модель комп'ютерної мережі для тестування алгоритмів маршрутизації трафіку. *Автоматика, комп'ютерно-інтегровані технології та проблеми енергоефективності в промисловості і сільському господарстві*: матеріали Міжнар. наук.-техн. конф., (Кропивницький, 10–11 листоп. 2022 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: Ексклюзив-Систем, 2022. С. 44–45.

5. Міхав В. В., Мелешко Є. В., Якименко М. С., Бащенко Д. В. Методи зберігання даних у рекомендаційних системах. *Цифрова трансформація суспільства*: тези доп. I Міжнар. наук.-практ. конф. (Кропивницький, 21–22 квіт. 2022 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2022. С. 44–45.

6. Міхав В. В., Мелешко Є. В., Якименко М. С. Дослідження можливостей графової бази даних Neo4j для застосування у розробці рекомендаційних систем соціальних мереж. *Комбінаторні конфігурації та їхні застосування*: тези доп. XXIII Міжнар. наук.-практ. семінару ім. А. Я. Петренюка, присвяч. 70-річчю ЛА НАУ (Запоріжжя–Кропивницький, 13–15 трав. 2021 р.) / за ред. Г. П. Донця. Кропивницький: Ексклюзив-Систем, 2021. С. 124–127.

7. Міхав В. В., Мелешко Є. В. Застосування методу delta-encoding для представлення даних рекомендаційної системи. *Інформаційна безпека та комп'ютерні технології*: тези доп. IV Міжнар. наук.-практ. конф. (Кропивницький, 15–16 квіт. 2021 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2021. С. 57.

8. Міхав В. В., Мелешко Є. В. Порівняння стратегій редагування бінарних діаграм рішень для роботи з графовими даними. *Інформація, комунікація, суспільство*: тези доп. 9-ої Міжнар. наук. конф. (Львів, 21–23 трав. 2020 р.) / М-во освіти і науки України, Нац. ун-т «Львівська політехніка». Львів: Вид-во Львів. політехніки, 2020. С. 17–18.

9. Міхав В. В., Мелешко Є. В. Метод оптимізації швидкодії бінарних діаграм рішень при представленні даних рекомендаційної системи. *Інформаційна безпека та інформаційні технології*: тези доп. II Міжнар. наук.-практ. конф. (Кропивницький, 2–3 квіт. 2020 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2020. С. 17.

10. Міхав В. В., Мелешко Є. В., Чабан О. О. Способи побудови рекомендаційних систем для соціальних мереж з врахуванням репутації користувачів. *Перспективні напрямки інформаційних і комп'ютерних систем та мереж, комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті*: тези доп. Міжнар. наук.-практ. конф. (Кропивницький, 13–14 лист. 2019 р.) / М-во освіти і науки України, Центральноукр. нац. техн. ун-т. Кропивницький: ЦНТУ, 2019. С. 86–87.

11. Міхав В. В. Алгоритми динамічного впорядкування змінних у впорядкованих бінарних діаграмах рішень. *Комбінаторні конфігурації та їх застосування*: тези доп. XVII Міжнар. наук.-практ. семінару (Кіровоград, 17–18 квіт. 2015 р.) / за ред. Г. П. Донця. Кіровоград: КНТУ, 2015. С. 75–78.